

# Strategies to develop a telemonitoring technology for machine tools via the World Wide Web

*I. Goncharenko, K. Mori, and N. Kasashima*

*Mechanical Engineering Laboratory, Agency of Industrial Science and Technology, Ministry of International Trade and Industry  
1-2 Namiki, Tsukuba, Ibaraki, 305 Japan, tel: +81-298-58-7074,  
fax: +81-298-58-7201, E-mail: mori@mel.go.jp*

## **Abstract**

The World Wide Web (WWW) can provide machining systems with remote monitoring capabilities through highly graphical user interfaces. As part of a feasibility study, an experimental system to check the availability of the current WWW protocols and languages for on-line machining data transfer and representation was developed. The system consists of a machine tool with PC-based NC, sensors, servers and terminal clients. The primary limitations of standard WWW techniques applied to the dynamic monitoring were found. A functionally distributed monitoring architecture to overcome these disadvantages is proposed. The present approach also provides recommendations for reconstructing complex user interfaces and virtual reality on the client side by utilizing a few of the machine tool parameters.

## **Keywords**

monitoring, machine tool, World Wide Web, graphical interface, virtual reality

## 1 INTRODUCTION

Modern manufacturing with geographically distributed systems may be remotely monitored and controlled. Such monitoring includes information collection and transfer from remote machine tools to control centers, data analysis both in automated and interactive modes. The information will enable the prediction of failures in vital parts of

the processes and, consequently, reduce the risk of high-cost losses and maintain overall system efficiently. Local area networks with information transmission speed of more than 300 kB/sec are being introduced in modern manufacturing plants. These environments provide manufacturers with an opportunity to utilize the on-line data obtained from their domestic and overseas manufacturing systems through the Internet.

The Internet has been developed as one of the most important world-wide computer communication mechanisms. In the last few years, WWW, initially designed by Berners-Lee (1992) and then supported by National Center for Supercomputing Applications (NCSA), has become an essential part of the Internet. It is a distributed information storage system, which provides graphical access to users through terminals (clients) to data sources (servers). The data described as documents in terms of the Hyper Text Markup Language (HTML), can contain references to other distributed documents by the Uniform Resource Locator (URL) addressing scheme. In addition to HTML, WWW is based on the HyperText Transfer Protocol (HTTP) providing an application level of the information transmission on the Web. At present, the WWW clients have ability to explore complex 3-dimensional scenes represented in the Virtual Reality Modeling Language (VRML) documents on servers (Pesce, 1995). The dynamic content of user graphical interfaces, corresponding to real time events, is yielded by either server-side CGI (Common Gateway Interface) techniques or client-side scripting applications, e.g., the Java applets (Ritchy, 1995).

The rapidly evolving WWW now offers interactive and automated on-line monitoring, data analysis, and control of remote machine tools. However, some limitations exist in the present WWW for these applications, and require improvements and new developments in protocols, languages, interfaces and techniques for the WWW. Machine tool monitoring also demands a specific system architecture.

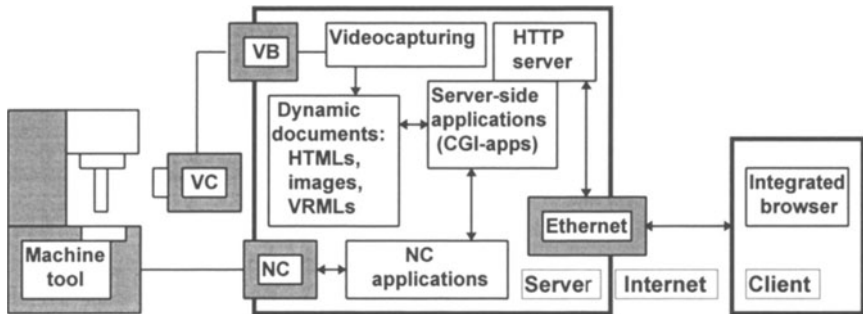
In this paper we discuss the restrictions of the commonly used WWW techniques in remote machine tool applications and control by constructing an experimental system. A distributed monitoring architecture is proposed to overcome the restrictions of extensive CGI methods and minimize the total amount of information transferred. Using this architecture, the HTML and VRML documents are reconstructed on the client side from a few on-line machine tool parameters. Consideration is also given to the WWW graphical representation of the parameters.

## 2 EXPERIMENTAL SYSTEM

In order to investigate the feasibility of the current WWW technologies for on-line machine data transfer, representation and control, a monitoring server-side system as shown in Figure 1 was developed.

The server-side system consists of Sakazaki SEC AE-61 3-axis milling machine controlled by Pentium PC/60 MHz computer with ST-8000 numerical controller (NC) from Techno Corp., Japan. The PC-based NC has its own RAM for CNC programs and serves the following functions; loading up to 12 CNC programs from PC disk drive into RAM and running one of them autonomously, directly controlling the machine tool, and

receiving a machine status. The status includes the real-time information about such parameters as the number of the running program, program step number, designated and actual x-,y-,z-coordinates of the tool, feed rate, spindle speed, axis motion status, and alarm time interval.



**Figure 1** Experimental system architecture.

The same PC receives video information from a CCD camera through the Intel Smart Video Recorder capture board (VB). The video board delivers luminance and chrominance data and forms color images in four sizes from 160x120 to 640x480 pixels. We selected the resolution of 320 by 240 to save space in the client display frame for other textual and graphical information. The 3Com EtherLink network adapter assures fast communication in our local area net between the monitoring server and clients.

The PC-based hardware was originally used for other purposes and determined the basic software components of the system. The capture board is controlled by Video for Windows with Indeo Video drivers, and access to the numerical controller is handled through the library of Microsoft C functions developed by Techno Corp. Thus Windows 3.1 was used as an operating system with Windows-based network software.

The video capturing program can store continuous video clips, but it exploits CPU resources exclusively and requires a huge amount of disk space. To avoid these factors and automate the capturing process, it was run in frame-by-frame mode under Windows Macrorecorder. A cycled macro, emulating fast key strokes, invokes video capturing and then our self-made application to provide a current video image.

Two applications serve the NC board. The first one provides an interface between server-side CGI programs and NC, receiving current machine tool parameters and sending control commands. The second autonomous application controls the machine tool in accordance with a pre-defined CNC program. One of the purposes of the CGI applications is to transform current on-line data and image into the documents in a WWW format and send them to client programs (browsers). These dynamic, "on-the-fly" documents contain random numbers in their URL names. This forces a browser to load the documents and

images every time. Otherwise, the documents with the previously executed URLs are extracted by a browser from its local cache.

From a client point of view, the system provides three functions at present:

- continuous graphical monitoring of the machine tool state;
- remote control of the machine tool;
- monitoring in virtual reality form.

Continuous monitoring means that a user can observe the current machine tool real image and parameters in a textual/graphical form. The user graphical interface is updated in time automatically. Otherwise, the remote control interface implies user interactive inputs by clicking a mouse to move or stop the machine tool. In the virtual reality interface the user can inspect "frozen", static artificial three-dimensional scenes corresponding to the real machine tool position at the moment of the interface invocation. Mac, PC, and Silicon Graphics Indigo<sup>2</sup> browsers were tested as clients of the server-side system, and the last one showed high capabilities of visualizing the virtual reality.

### 3 ON-LINE DATA REPRESENTATION ON THE WEB AND DATA SIZES

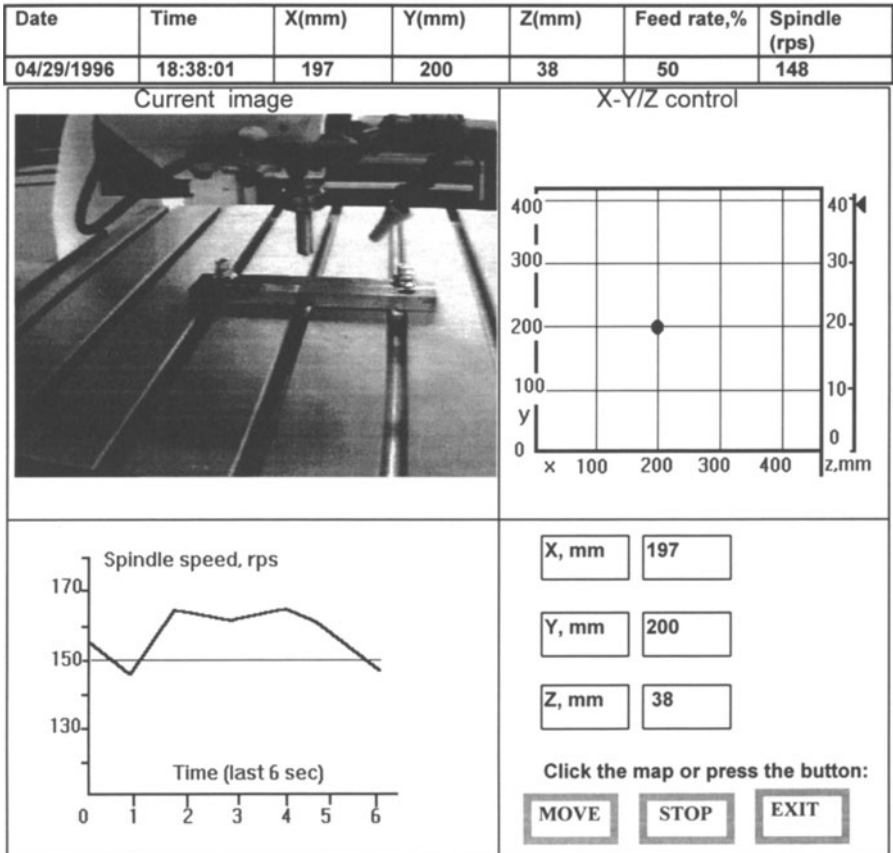
Information about the current state of machining processes can be obtained through the Numerical Controller and various sensors. Their outputs are: tool coordinates, feed rate, cutting force, spindle rotation speed, vibration level, acoustic emission, temperature, optical images, etc. All these data are essentially continuous and measured in real-time,  $t$ . After digitizing they can be depicted as a finite set of discrete one-dimensional functions of  $t$  or multi-dimensional functions of  $t$  and other parameters (coordinates, frequencies).

But how can the data be represented on the WWW? It depends on a successfully designed intuitive user interface, because WWW browsers only graphically interpret server-side HTMLs/VRMLs. Originally, the Web was created to retrieve technical documents. Thus, the simplest way is to represent the on-line data in the form of plain text, formatted text (hypertext) and tables, utilizing current time and date together with measured parameters transferred from a server.

Compared to tables, graphs are more human-friendly for presenting time-dependent functions. Developing WWW for increasing the graphical capacities, NCSA added in-line images into WWW documents in several graphical formats. One of the formats, GIF (Graphics Interchange Format) is well suited to the representation of plots, graphs and pictorials. GIF is based upon the lossless compression algorithm (Ziv, Lempel, 1978) designed especially for raw data with repeated substrings such as raster images of plots. Otherwise, the JPEG (Joint Photographic Expert Group) format with "lossy" compression algorithm provides much better results for natural true-color and gray-scale images (Aravind et al, 1989). It is important to choose a specific compression method for reducing the transferred image data size without serious losses in quality of "visibility" .

Figure 2 shows the user interface for remote monitoring and control of the experimental system. The interface contains on-line parameters in tabular form (1.6 kB of HTML document), JPEG video image of the machine tool (8.5 kB), GIF image of spindle speed plot (1.6 kB), and schematic GIF image of the tool position for point-and-click control (2.8 kB).

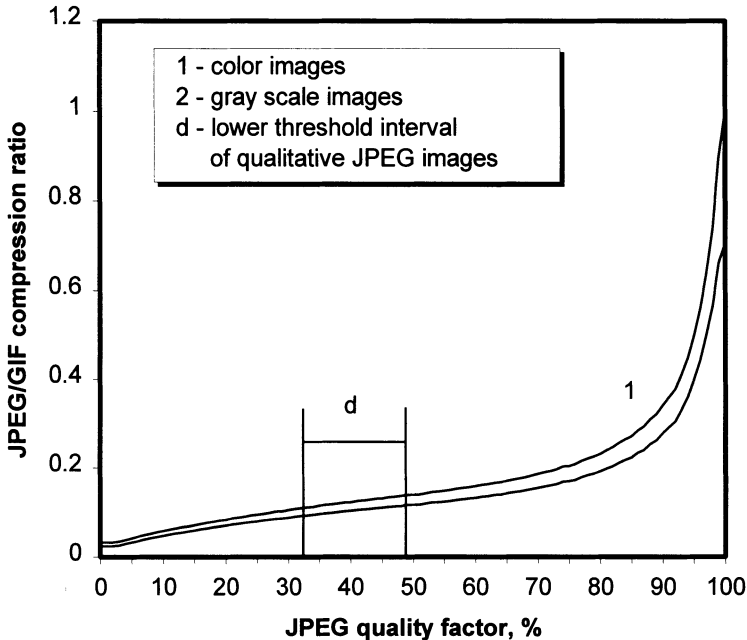
### Remote Control



**Figure 2** Graphical user interface (remote control).

The GIF compression ratio indicates complexity of an image and depends on its nature - texture, repetitiveness of pixels, level of details and noise. Trying to reduce data sizes of photographic images, several image sequences of typical operating scenes were tested. It

was found that the JPEG pictures have 7-10 times less size than comparable GIF images, as shown in Figure 3. The lower threshold of JPEG quality factor was estimated subjectively when independent experimenters detected the difference between original and compressed images. The figure also displays the efficiency of JPEG format for color images. When JPEG compression is applied to artificial graphs and plots, it proves inefficient and causes an erosion of drawn objects.



**Figure 3** Averaged JPEG/GIF compression efficiency for typical operating scenes.

In the system, the process of monitoring enables a remote operator to observe the machine tool state and control it through an "updating-in-time" graphical interface. The graphical information is arranged in a display frame as shown in Figure 2. At a defined interval of the updating time, for example 1 sec, total data size can be estimated from Table 1. The above interface without JPEG image requires 6 kB to be transferred.

Besides JPEGs and GIFs, other WWW data with certain MIME (Multipurpose Internet Mail Extensions) media types (Borenstain, Freed, 1989) can be retrieved by the client. In our case, even simple VRML 1.0 document describing one static 3D operating scene requires about 5 kB. The size of VRML documents strongly depends on the degree of comprehensibility of the scene described.

Therefore, to represent in graphical form a few on-line parameters initially stored in 20-30 bytes on the server side, it is required to transfer to the client 5-10 kB of documents! In this case, the bandwidth of the Internet is a serious restriction for monitoring. For example, the bandwidth of our local area network varies from less than 1 to more than 300 kB/sec, depending on a server/client location and time of day. For some clients the latency between real event in the machine tool site and the appearance of corresponding graphical representation exceeds 10 seconds. This latency is too large to control the machine tool adequately or stop it in case of emergency.

**Table 1** Types of on-line data and data sizes (\* - 1sec playback)

<i>Type</i>	<i>MIME media type</i>	<i>Data size, kB</i>
Text, tables	text/plain/html	0.5..5
Graphs, plots, pictorials	image/gif	2..5
Images	image/jpeg	5..20
Audio	audio/basic/wav (*)	5..10
Video	video/mpeg/msvideo (*)	50..1000
VRML data	x-world/x-vrm	5..500

If it is assumed that the frame is accompanied by real-time audio/video information transferred through the same network, the data size per 1sec increases dramatically (Table 1). At present popular browsers on various platforms include internal tools to playback audio/video multimedia files in "real-time" simultaneously with receiving the files. On the server side, the files must be prepared and described in HTML *in advance*, so there is not a continuous stream of information required for monitoring. Otherwise, InPerson videoconferencing technology, developed by SGI and promoted through NetManage for PCs, provides useful properties for inclusion in our system. The estimated Internet bandwidth required in both cases is at least 50 kB/sec.

Thus, the first strategy for successful "real-time" monitoring is to reduce the total amount of data transferred through the network. The limitation in bandwidth is the first restriction on monitoring via the WWW, but not the last. In our fast local network with a bandwidth of more than 300 kB/sec, three types of user interfaces were tested. With the first, including only on-line parameters in plain text format, it was possible to update the information 10 times per second. When two GIF images and then a JPEG image as shown in Figure 2 were sequentially added, the updating rate was reduced to 2 and 4 seconds respectively. Moreover, in all cases the updating rate was not time-uniform. The reasons of these disadvantages are considered next.

#### 4 DISADVANTAGES OF SERVER-SIDE SCRIPTING

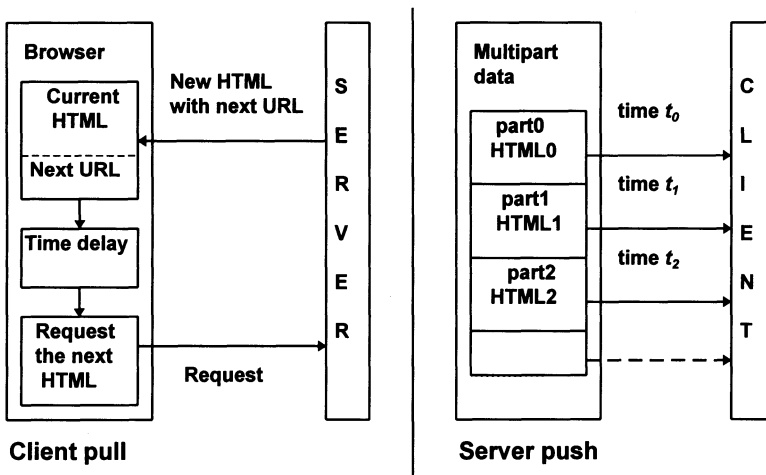
##### *Time non-uniformity of the monitoring*

Every new cycle of monitoring requires generation of the dynamic WWW documents corresponding to current operating parameters. The client-side control also is possible by

transferring the user inputs to controlling programs existing in the server-side system. As a WWW convention, these properties are maintained by the NCSA CGI technique.

CGI is an interface between client and server for running external programs, or gateways, under an information server. The gateways can be ordinary executable programs that provide standard output in the form of scripts. These scripts contain at least the MIME type definition of transferred data and the WWW document itself or its location. Commonly, input information for the gateways is passed in an interactive mode; a user of the client browser fills and submits FORM and ISINDEX requests of his current HTML document. The requests are passed through CGI and handled by a server. Started on the monitoring server, the gateways receive and process on-line data, control the machine tool, create new WWW documents and then output the appropriate scripts for the server. In such a manner several authors developed robot systems teleoperated through the WWW, for example, Goldberg of the University of Southern California (Goldberg et al, 1995).

This method is satisfactory for the interactive remote control and generation of current virtual reality scene, but the continuous telemonitoring requires the documents to be updated automatically on a periodic basis. We tested two complementary WWW mechanisms providing the automatic document loading, namely, "server push" and "client pull". In Figure 4 both techniques are shown schematically.



**Figure 4** CGI techniques for continuous monitoring.

In the "client pull" method, the server sends down a special directive followed by main document body, for example;

```
<META HTTP-EQUIV=REFRESH CONTENT= "5; URL= ttp://host/NewRandom URL">
Document body
```



The browser will fully display this document, then wait 5 seconds and call a new gateway defined by the URL. In this technique, a client-server HTTP connection is held open only for transferring a current portion of data. Otherwise, in the "server push" method the connection remains open for an indefinite time period (Figure 4). The data transfer is fully controlled by the server. Initially, the server sends a special MIME content type descriptor and boundary terminator, informing the browser when every portion of data will be finished. Then CGI scripts force the server to send current documents, HTML0, HTML1, HTML2 at times  $t_0$ ,  $t_1$ ,  $t_2$  respectively. When the browser receives the terminator it waits for the next document.

Various experiments were carried out with the experimental system, and it was found that these server-side scripting techniques have the disadvantage of highly non-uniform playback on client side. In case of "server push", it is possible to tune the time difference  $t_1 - t_0 = t_2 - t_1 = \dots$  to a constant value from server side only, but the browsers do not synchronize multipart MIME data. In "client pull", the time uniformity also cannot be defined by the META tag because a different time required for preparation of dynamic documents and images on the server. In addition, a data transfer time is unpredictable. For example, the updating time of the user interface, shown in Figure 2, was varied from 3 to 8 seconds.

Therefore, the next strategy for telemonitoring is to use a synchronizing mechanism on the client side, which will be discussed in Chapter 6.

#### *Extensive nature of gateways*

CGI applications use much processing time, preparing dynamic documents in WWW format. The graphical representation of on-line information adds to CGIs functions such as drawing and GIF compression. The gateways also generate random URL names for each new document. The CGI scripting describes that the gateways write new HTML/VRML documents directly on standard output or prepare them using pattern files.

When several clients monitor the machine tool simultaneously, the number of gateways increases and it can overload the server. In order to discharge the monitoring computer, the functions of dynamic document preparation should be passed from server to client side.

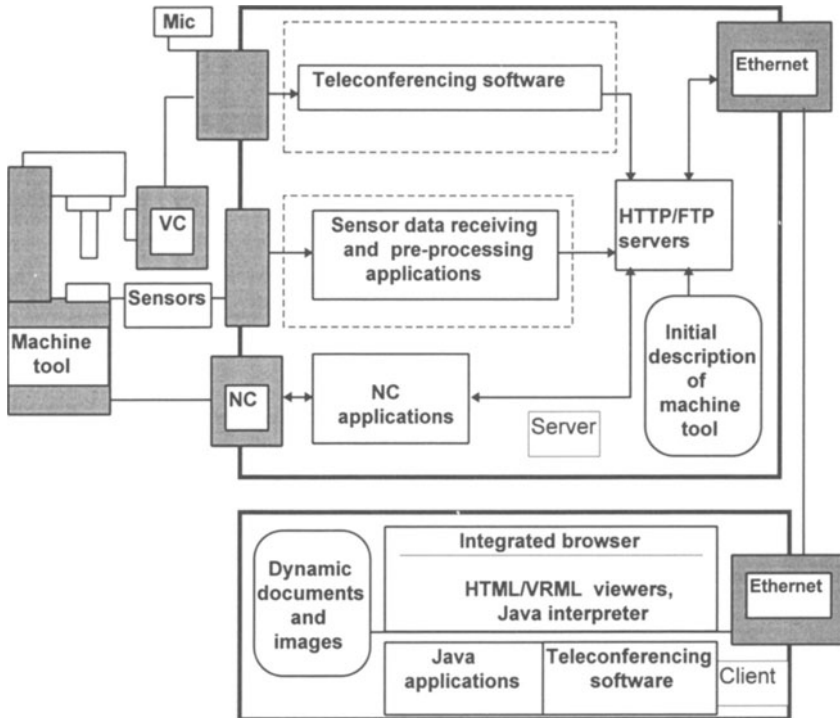
## 5 CLIENT-SIDE DYNAMIC INTERFACE CONTROL AND VRML

Actually, the above mentioned strategies of the telemonitoring are interrelated. A client-side control may result in the system flexibility and minimal data transfer. The successful solution of building a monitoring system is based on optimal distribution of server-client functions. We propose such distributed system architecture (Figure 5).

The server-side computer serves the functions of the machine tool control, on-line data receiving and transfer to clients in a compact file format. If the Internet bandwidth is adequate, teleconferencing can be added.

All static parts (HTML patterns, decorating images) of the graphical interface associated with monitored operating parameters are pre-loaded from an initial description of the machine tool (MTID) to the client computer. MTID contains algorithms to create the

dynamic interface from these static parts. Java portable code is used to yield a platform independence of the algorithms. On the client computer, the Java code receives information from the machine tool through its Network Class Library (Ritchy, 1995) and continuously updates the user interface. This Java code controls the time-uniformity of interface updating.



**Figure 5** Distributed server-client architecture.

MTID additionally includes VRML geometrical descriptions of machine parts, CNC programs of the machine tool and NC-control algorithms both to simulate machining processes in off-line mode and calculate on-line VRML scenes. At present, in the same manner we utilize on-line data with the CGI method to reconstruct virtual scenes. For example, in the server, three parts of the machine tool, corresponding to each coordinate axis, are described geometricaly in VRML as follows:

```
#VRML V1.0 ascii
Separator {
  .....
  Separator { #part1
```

```

    Translation { translation x1 y1 z1 } Rotation { rotation u1 v1 w1 }
    ..... }
Separator { #part2
    Translation { translation x2 y2 z2 } Rotation { rotation u2 v2 w2 }
    ..... }
Separator { #part3
    Translation { translation x3 y3 z3 } Rotation { rotation u3 v3 w3 }
    ..... }
}

```

Receiving the current on-line x-,y-,z-coordinates of the machine tool and CNC program step number, gateway calculates new x1, y1,..., w3 for the pattern and replaces them. Then the modified VRML is transferred to the client browser.

For the new architecture, there are no reasons to pass whole VRML documents every time, because the pre-loaded pattern is modified on the client side in accordance with the on-line coordinates transferred. It is possible to use this advantage by reconstructing VRML display images into user interfaces instead of real video images of the machine tool. Moreover, the initial VRML pattern can provide more level of details than the real image.

At present our VRML scenes are static, so users only examine a "frozen world" of the machine tool. In future applications, Cosmo tools from Silicon Graphics for VRML 2.0 will be used. This software provides moving virtual reality by Java scripting technique, which is well suited for our distributed system.

## 6 SUMMARY AND CONCLUSION

Future manufacturing system will need network communication between world-wide distributed manufacturing machines and information processing subsystems. This communication must serve the dual function of telemonitoring and adequate machine tool control according to the monitored information. At present, the World Wide Web as an information system on the Internet, has two remarkable features: (1) it provides an access to distributed and cross-linked data sources, and (2) it gives a graphical user interface. Our application field confirms that WWW technologies have evolved in the right direction.

With the experimental system developed, the possibility of transferring and representing information about a current machining state has been tested. The experiments confirmed that dynamic on-line textual, graphical and visual information can be retrieved on a continuous basis by using gateways, "server push" and "client pull" mechanisms. This standard CGI technique was found to be adequate for remote control. However, for continuous monitoring it has several disadvantages; consuming vast server resources, increasing the volume of non-relevant information transferred, and not providing time-uniformity of updating the user interface.

It was also found that the low bandwidth of the Internet is a basic restriction for monitoring applications. With the framework of CGI, successful monitoring is possible only for minimal graphical interfaces. Real time audio and video information requires at

least 30-50 kB/s of the bandwidth. Additionally, such synchronizing tools are required for "live" data streaming, as a real time protocol or InPerson teleconferencing.

To overcome these limitations, a functionally distributed architecture of the telemonitoring has been proposed. This approach reduces the total amount of transferred data and retains the rich graphical representation of machine tool operating state. It is achieved by utilizing only a few of the on-line parameters to generate a display frame (including artificial images and moving virtual reality) on the client computer.

The main results of this study for the purposes of on-line machine tool telemonitoring and control are as follows.

(1) After testing several graphical interfaces, the main restrictions of the telemonitoring via WWW were found. All the restrictions originate from the extensive nature of the standard method for generating dynamic documents. Using this method only the simplest user interfaces are available for continuous monitoring, but remote control is still possible.

(2) A distributed client-server architecture minimizing the amount of transferred data and concentrating only on on-line data was proposed. The new approach can get around the limitations by the use of the latest WWW tools.

## 7 ACKNOWLEDGMENT

The authors wish to thank Dr. T. Kojima of the Mechanical Engineering Laboratory for initiating our interest in application of the VRML for the machine tool monitoring.

## 8 REFERENCES

- Aravind, R., Cash, G.L., and Worth, J.P. (1989) On implementing the JPEG still-picture compression algorithm, in *Proceedings of SPIE Conference Visual Communications and Image Processing*, IV, 799-808. SPIE, Bellingham, WA.
- Berners-Lee, T., Cailliau R., Groff, J.-F., and Pollerman B. (1992) World-wide Web: The information universe. *Electronic Networking: Research, Applications and Policy*, **2(1)**, 52-58. Meckler publications. Westport, CT.
- Borenstain, N. and Freed., N. (1993) MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for specifying and describing the format of Internet message bodies. Internet RFC 1521.
- Goldberg, K., Masha, M., Gentner, S., Rothenberg, N., Sutter, C., and Weigley, J. (1995) Desktop teleoperation via the World Wide Web, in *IEEE International Conference on Robotics and Automation*, 654-659. Nagoya, Japan.
- Pesce, M. (1995) VRML. *Browsing and building cyberspace*. New Riders Publishing. IN.
- Ritchy, T. (1995) Java! New Riders Publishing. IN.
- Ziv, J. and Lempel, A. (1978) Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory*, **24(5)**, 530-536.