DEVELOPMENT OF VISUAL 3D VIRTUAL ENVIRONMENT FOR CONTROL SOFTWARE

Michitaka Hirose, Takeshi Myoi, Haruo Amari, Kohei Inamura, and Lawrence Stark
University of Tokyo and Tokyo Electric Power Co., Tokyo, JAPAN
and University of California, Berkeley California

Virtual environments for software visualization may enable complex programs to be created and maintained. A typical application might be for control of regional electric power systems. As these encompass broader computer networks than ever, construction of such systems becomes very difficult. Conventional text-oriented environments are useful in programming individual processors. However, they are obviously insufficient to program a large and complicated system, that includes large numbers of computers connected to each other; such programming is called "programming in the large."

As a solution for this problem, the authors are developing a graphic programming environment wherein one can visualize complicated software in virtual 3D world. One of the major features of the environment is the 3D representation of concurrent process. 3D representation is used to supply both network-wide interprocess programming capability (capability for "programming in the large") and real-time programming capability. The authors' idea is to fuse both the block diagram (which is useful to check relationship among large number of processes or processors) and the time chart (which is useful to check precise timing for synchronization) into a single 3D space. The 3D representation gives us a capability for direct and intuitive planning or understanding of complicated relationship among many concurrent processes.

To realize the 3D representation, a technology to enable easy handling of virtual 3D object is a definite necessity. Using a stereo display system and a gesture input device (VPL DataGlove), our prototype of the virtual workstation has been implemented. The workstation can supply the "sensation" of the virtual 3D space to a programmer. Software for the 3D programming environment is implemented on the workstation. According to preliminary assessments, a 50% reduction of programming effort is achieved by using the virtual 3D environment. The authors expect that the 3D environment has considerable potential in the field of software engineering.

PROGRAMMING IN THE LARGE

The authors are involved in the design project for future power supply system in Tokyo. The goal of the project is to have the entire power distributing system controlled automatically by a vast computer networking system. In order to realize that goal, we need to find ways to solve the problems which arise in developing such a large and complicated software scheme.

Two basic types of problems associated with the different levels of the networking environment can be considered. One type of question might be: "How do I program each computer?" (This is called programming in the small). Another mights be: "How do I coordinate many computers?." (This is called programming in the large.)

Since the programming environment itself is built on "algorithmic" or "logical" characteristics, it is convenient to use text representation. With text, we can eliminate much ambiguity. However, text alone becomes limitingly insufficient in the case of "programming in the large" where the most important factor in the field concerns total configuration design rather than detailed logic design. Thus a more global methodology is needed to correct for faults in the overall system design, even though each individual subsystem may be correctly designed.

In "programming in the large," the authors believe that visual representations play an important role in the intuitive understanding of software and a combination of text and visual environments will provide the best solution.

CONCEPT OF 3D VISUAL REPRESENTATION

Often, a network diagram is given as a 2D representation, known as block diagram. However, if the description of a real-time control program is required, such as in the synchronization of several control processes, the time dimension should also be taken into consideration.

Fig. 1 shows the concept of the 3D visual representation. If we observe the representation along the X axis, it will be a conventional block diagram. The axis normal to the X axis is a time chart. Using this 3D representation, the programmer can intuitively grasp the state of concurrent process.

In the 3D visual representation, the geometrical shape used to represent program flow of successive message passing is also important. The shape itself indicates the type of information exchange. Sometimes the programmer can handle the software entirely based on its shape. The greater the number of processes, the greater the advantages of 3D representation, presumably due to characteristics of the human cognitive processes.

IMPLEMENTATION

Several years ago, such an idea was unrealizable due to the limitations of computer capabilities. To implement 3D representations, we need a new technology able to incorporates real-time 3D animation, stereo displays, 3D input devices, and other aspects of the virtual 3D work-space. To generate a 3D image, the authors used Stereographics CrystalEyes System, (a flickerless 3D CRT by high speed LC (120 Hz Liquid Crystal) shutter). By adding a head movement detector (Polhemus 3D tracker) to that system, a partial "look-around" effect can be implemented. As a gestural input device, VPL DataGlove is used.

Fig. 2 shows the current hardware configuration of the virtual 3D workstation. The system includes two sub-workstations. One of them (an HP9000 SRX) is used for generating the virtual 3D image and handling I/O devices. Another (a Sun 3) is used for text handling. In other words, the first one is for "programming in the large," and the second is for "programming in the small."

Figure 3 shows an example of the visual programming environment. The message flow between processes is displayed in virtual 3D space. One small box represents a computing object (sometimes it can be a process, sometimes it can be a processor).

Semi-tranparent box represents a carrier of computing objects (it can be a stand-alone computer box).

Using a virtual hand which works as 3D pointer and a kind of "action menu" which detects gestures, programmer can handle virtual objects (such as "objects" and "messages") in the virtual 3D world. Currently, the environment has two modes, called "world mode" and "succession mode." World mode is mainly for observing whole shapes of the software. Succession mode is used for editing messages passing among objects. The configuration is still preliminary. Further investigation is needed concerning the design of object handling in 3D virtual space.

Tools included in the environment are as follows:

1. Virtual Measure/Ruler: Tool to measure the exact relationship among virtual objects.
2. Critical Path Finder: Tool to find and display the critical path which determines the total network throughput for a given network task.
3. Network Simulator: Tool to simulate message passing and data processing on each computer.
4. Network Planner: Tool to update information about the network configuration. This tool works with Network Simulator.

ASSESSMENTS AND CONCLUSION

As a preliminary assessment of the 3D environment, several multiprocessor programming tasks were assigned to subjects. The target system is supposed to be 3-4 networked personal computers. Completion time with and without the environment was measured. Roughly speaking, completion time with the 3D environment was shorter than that of the conventional methodology. In addition, using 3D lowers the variance, indicating that the environment reduces error.

The authors believe that the concept of 3D visual representations has proven to be effective even in the simple environment already implemented. A far greater impact can be expected in a more sophisticated environment as the most case of software environment. For example, Smalltalk80 is highly valued for its concept of "object oriented" programming, but its environment is valued much more than the concept itself. The authors consider this kind of practical effort is indispensable. From a practical point of view, however, much effort is needed to prepare and accumulate effective tools for this new environment.

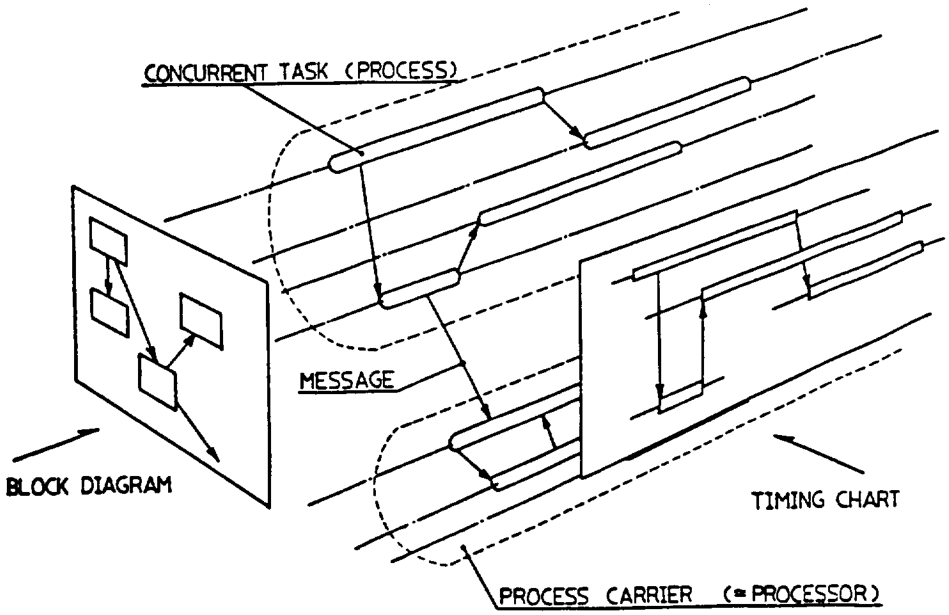

Figure 1. Concept of 3D visual representation.

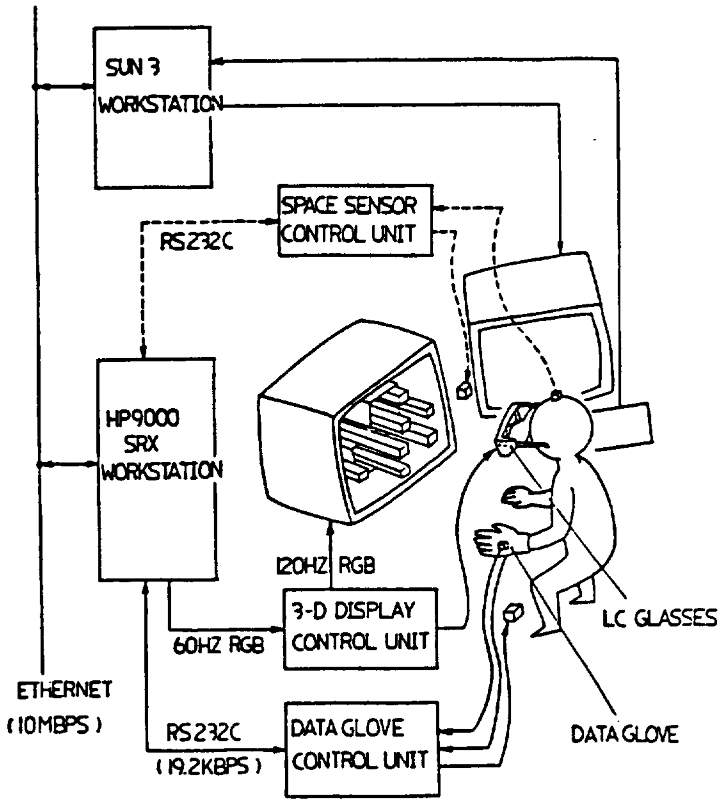

Figure 2. Hardware configuration of the virtual 3D workstation.

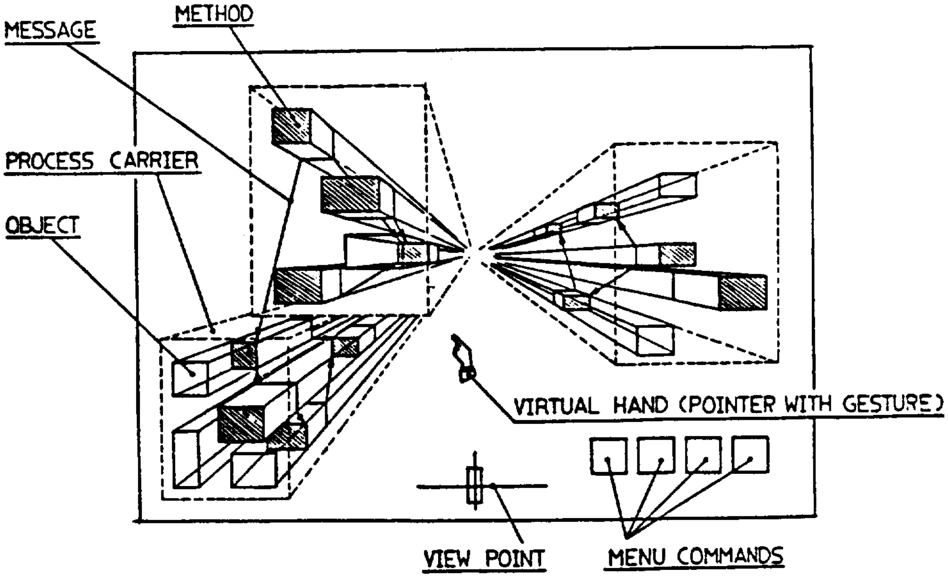

Figure 3. Example of visual 3D programming environment.