N94- 27369

AN ARTIFICIAL REALITY ENVIRONMENT FOR REMOTE FACTORY CONTROL AND MONITORING

Chuck Kosta and Patrick D. Krolak
University of Massachusettes Lowell
Lowell, Massachusettes

Abstract

Work has begun on the merger of two well known systems, VEOS (HITLab) and CLIPS (NASA). In the recent past the University of Massachusetts Lowell developed a parallel version of NASA CLIPS, called P-CLIPS. This modification allows users to create smaller expert systems which are able to communicate with each other to jointly solve problems.

With the merger of a VEOS message system, PCLIPS-V can now act as a group of entities working within VEOS. To display the 3D virtual world we have been using a graphics package called HOOPS, from Ithaca Software. The artificial reality environment we have set up contains actors and objects as found in our Lincoln Logs Factory of the Future project. The environment allows us to view and control the objects within the virtual world. All communication between the separate CLIPS expert systems is done through VEOS.

A graphical renderer generates camera views on X-Windows devices, Head Mounted Devices are not required. This allows more people to make use of this technology. We are experimenting with different types of virtual vehicles to give the user a sense that he or she is actually moving around inside the factory looking ahead through windows and virtual monitors.

1 Introduction

This work represents one effort to produce technology which will allow the region and the nation to compete in the world market. It has centered upon flexible manufacturing and intelligent workcell control. The artificial reality environment currently under construction will demonstrate the current and future applications for artificial reality tools in the factory.

2 Historical Perspective

In the early days of the industrial reveolution it was possible, if not required, for machinists to work within feet of the machines s/he were to control. Machines produced their own information in forms like sounds, odors, and "the feel" of the working unit.

Later, the central control room came into being. Here one could find whole rooms full of readouts, charts, dails, and warning bells. For some people, it was difficult to be away from their machines, even these few yards. No longer were there noises and odors to be had. Often it took a new generation of employees to learn to use the control room gadgets in a productive manner.

As the number of automated machines increased, fewer controls could be kept in a single control room. In todays, factories controls are being distributed in a clustering manner. These machine clusters then report in a "control room"-like manner to centralized monitors and strip charts which allow for recording and monitoring. Programmable controllers handle most of this reporting function. IBM PCs and clones are being used as front ends to these distributed control sites. Graphs and visual programming languages (ladder logic and flow diagrams) are being used to control these machines. Control information can be downloaded from the remote control rooms as well as at the local machine.

In this work we propose that three dimensional graphics can be used recreate gadgets such as toggle buttons, numeric readouts, slider controls, and other controls. One can now take the control room to the person, instead of the person going to the control room. In fact many people can manipulate and view the same control panel at the same time.

In addition to creating the control panels for the factory, an artificial reality environment can also reproduce the physical machines and objects. One such example is the ARKola Simulated Bottling Plant developed by [GSO91]. In this artificial world multiple people manage different parts of the factory and interact with one another.

3 The Virtual Control Panel

In addition to generating the artificial world, it is also possible to insert knowledge into a scene by utilizing visualization techniques. First, visual mapping parameters are inserted into the rendering pipeline. Second, floating text is used as Heads-Up-Data (HUD) on top of the rendered objects.

To get around in the factory (A.R.) we are exploring different models. At present we are using a monitoring camera paradigm. The viewstation that we generate on the screen contains one main simulated monitor and three smaller monitors. The camera that is "patched-in" to the main monitor location can be manipulated be the controls on the viewstation including: Pan, Orbit and Dolly camera options. The three cameras can be looking any where in the artificial world, there do need to be different views of the same work area.

In future experiments we are considering virtual vehicle for traveling around the factory. These controls would allow for objects like a golf cart, a mobile robot, and a UFO. These virtual vehicles would be used to let the user enter a desire location into a piloted vehicle. Currently, we have attached a simulated camera to the top of one of the simulated mobile robot pickup arms. As the robot moves around the factory you can watch where it travels and control the direction of the camera on the pickup arm.

4 The Virtual Factory of the Future

The artificial reality consists of artificial entities that share a portion of their knowledge base. This is then rendered by one or more of the entities using a 3D object oriented graphics system called HOOPS. HOOPS is a rendering and input system developed by Ithaca Software, Inc. HOOPS allows for both presentation and mouse based input. We use the mouse mainly for picking and menu options. However, you could create any imaginable widget under mouse control.

The artificial world will contain full three dimensional objects (either boxlike or actual CAD descriptions). These objects will be placed in the artificial world in a similar arrangement for each person in the environment. This allows the spatial relationships to be shared with others. However, the views of the world are up to the individual, tailoring the monitor-like objects and Heads-Up-Data.

In the virtual factory of the future there will be teams of professionals. Each participant will share, form separate locations, the controls of the factory floor. Factories in one part of the world can be monitored and controlled from another part of the world. It will even be possible to meet at the same (syncronous) time, and jointly solve an engineering or manufacturing problem. The virtual factory of the future will still contain workers. There will be local technical repair teams who will be cordinating with others via Artificial Reality, Video Conference, or some other highbandwidth communication link.

5 The Lincoln Logs Factory of the Future

The Lincoln Log Factory of the Future was designed to be highly autonomous, ideally, needing minimal help from the user. The goal is to use multiple expert systems in a cooperative communication environment to develop an intelligent manufacturing environment. The system will control multiple robots, parts feeders, vision requirements, and a materials handling interface. The current model of the factory of the future utilizes an integrated Computer Aided Engineering (CAE) environment. The computer aided design (CAD) package has knowledge of structural requirements and part constraints. It requires the user to select parts which can actually be placed. The intelligent CAD system creates a work order, represented by structured English sentences, sent to the factory scheduling software.

5.1 THE FACTORY

The factory software is made up of multiple interdependent modules running individually. Included in this model is the opportunity to replace modules with others of equivalent functionality. Chief amongst these interchangeable modules is the simulator. The simulator process can present a three-dimensional view of the workcell. Physical properties such as gravity and friction are also simulated within the graphical environment. The modules which make up the factory software include:

5.2 RECEIVER MODULE

This module is used to monitor external input into the workcell, which it redirects to the appropriate process(es). The external input can come from one of three sources. First, a virtual control panel, described above. Second, an operator console which consists of a process containing graphical information regarding the robot statuses. The final source is a higher level scheduler, called a POD scheduler. The POD scheduler is responsible for controlling multiple workcells.

When the RECEIVER process receives a startup message from an input source, it creates the other processes in the system. The workcell configuration message is included in the startup message. The configuration is passed along to the other processes in the system, once they have started.

5.3 SCHEDULER MODULE

This process is used to assign assembly tasks to the robots. It reads the assembly instructions from the CAD system's English sentence file. These instructions are used for assigning tasks to the robots. The order in which these tasks are carried out is not specified. The scheduler determine the optimum order in which to carry out the tasks, and it constantly updates that order, depending upon robot load, external input, and mechanical errors.

When the scheduler receives a task request from one of the robot processes, it examines the current state of the house. Along with the parts that could be added next. It then determines the next optimal parts to be added to the house. The next optimal part is determined by a combination of dynamic load balancing, and collision avoidance scheduling. Dynamic load balancing is achieved by placing critical parts into the house at the earliest point possible.

5.4 ROBOT MODULES

There are two robots per workcell. When a robot starts up, it sends a request to the SCHEDULER, for an assembly task. The SCHEDULER assigns the optimal task to the ROBOT process. The ROBOT module must then issue a feed command to the appropriate feeder, move the robot arm to the feeder, grasp the part, and move it out of the parts feeder. The robot then moves the part to the edge of the workspace, and issues a request for access to the workspace, to the PREVENTER process.

Once granted access to the workspace, the robot moves the robot arm to place the part, and releases it. The robot then moves out of the workspace, and informs the PREVENTER of its action. If vision inspection is enabled, the process sends an inspection request message to the VISION system and waits for a response. An error in the part placement will cause another request to obtain the workspace. The robot then returns to the place where it released the log, and shifts the log into the correct position. Another inspection request is made to verify placement.

5.5 VISION MODULE

This process provides the communication connection to the vision system. When a ROBOT requires a vision function, a corresponding message to the VISION process is sent. The message is forwarded via serial line to the vision system. The VISION process waits until it receives the feedback from the vision system, which it passes along to the requesting ROBOT process. If the vision system were to become disabled, the VISION process would recognize the problem, and report it to the ROBOT and SCHEDULER processes. The vision system is monitored for restoration, and if it occurs, the information is passed to the other processes.

5.6 PREVENTER MODULE

There is always the possibility of the robots colliding in a multiple robot workcell. There are many ways of preventing his situation. One is to enforce mutual exclusion of the critical area. The PREVENTER process performs collision prevention by calculating where each robot arm, gripper, and part will be located during placement. If a collision is detected, the PREVENTER will enforce mutual exclusion of the workspace, otherwise, both robots can access the workspace simultaneously.

5.7 OPERATOR CONSOLE PROCESS

This is a process running on a computer workstation. It receives the workcell output from the DISPLAY process. It has a live video window right on the monitor, enabling the operator to see what is actually taking place in the workcell. The operator has complete control of the workcell from the console. This includes startup, reconfiguration, and shutdown capability. The operator has the option of adjusting the following functions in the workcell: Vision inspection, vision placement, operator mode, and compliant movement. The operator may also shut down any of the parts feeders, or either of the robots.

5.8 COMMUNICATION MODULE

The communication module is the heart of the system. It must maintain a robust interface to all the other modules and subsystems. It is important that the communication be done in a manner transparent to the programming environment. This allows for ease of use and easy replacement of code. Soon, some of the modules will be moved to another host - this will be facilitated when the communication module can talk across hosts without any subsystem knowing the difference. The communication module is being developed to send messages to cooperating subsystems in CLIPS, and to other mailbox-type programs via a C language interface with the underlying operating system.

6 The VCLIPS Architecture

We have merged portions of VEOS from the HITLab in Seattle, with our own coarse-grain parallelism extensions to Clips called PClips (Parallel Clips). It will be possible to receive both PClips communications and VEOS messages. To accomplish this merger of both PClips and VEOS, we removed the XLisp level from the distributed VEOS code. Simply using the "talk" layer of the VEOS environment we can send PClips messages back and forth. VEOS has design accomplishments similar to PClips in that they both have an entity based design and seek multiplatform capabilities. We chose VEOS because it has the potential to become a de facto standard within the research community.

The combination of VEOS and PClips will allow us to develop knowledge bases with smaller rule sets, yet still allow the expert systems to interact to solve group problems. Also, Clips is growing in usage due to its cost and continued development efforts by NASA, making it an excellent base to build upon.

7 Future Research Considerations

Further research will be done in the areas of artificial reality-based user interfaces, virtual vehicles which can be used to move around in the artificial worlds and realtime control of physical objects from within the artificial reality. Additionally, we are seeking industrial partners who are interested in experimenting with artificial reality based monitoring of an actual factory floor.

One new project will be using the artificial world to train a neural network. The neural net will then be inserted into a real mobile robot and used to recognize intersection patterns that it had learned. In this work the artificial reality will contain a description of an office building. The simulated robot will continually roam the simulated office trying to learn the different locations. The neural net will then be loaded into the actual robot to test whether it can actually determine where it is based on the different sensory input it receives from the real world.

We also hope to connect to other virtual world based research which may be interconnected on the internet. Providing object translators or visualization mappings for different VR and AR systems in real time.

References

[CJK+92] Christopher Codella, Reza Jalili, Lawrence Koved, J. Bryan Lewis, Daniel T. Ling, James S. Lipscomb, David Rabenhorst, Chu P. Wang, and Greg Turk. Interactive simulation in a mulit-person virtual world. In *Proceedings ACM SIGCHI '92*, pages 329–334, 1992. (IBM Watson R.C., Veridical User Envs.).

[CRM91] Stuart K. Card, George G. Robertson, and Jock D. Mackinlay. The information visualizer, an information workspace. In *Proceedings ACM SIGCHI '91*, pages 181–188, 1991. (Card.PARC@Xerox.com).

[ES90] Margaret A. Ellis and Bjarne Stroustrup. *The Annotated C++ Reference Manual.* DAddison-Wesley, Reading, MA, 1990.

[FB89] Jay Fenton and Kent Beck. "playground": An object-oriented simulation system with agents rules for children of all ages. In *Proceedings OOPSLA '89*, pages 123–136, October 1989. (Apple Vivarium Project).

[GA90] Michael Girard and Susan Amkraut. "eurhythmy": Concept and process. *Journal of Visualization and Computer Animation*, Vol. 1, No. 1, August 1990.

[GSO91] William W. Gaver, Randall B. Smith, and Tim O'Shea. Effective sounds in complex systems: The arkola simulation. In *Proceedings of HCI '91*, pages 85–90, 1991.

[KL90] Dennis Kafura and Keung Hae Lee. "act++": Building a concurrent c++ with actors. *Journal of Object Oriented Programming*, pages 25–37, May/June 1990.

[KWN90] Dennis Kafura, Doug Washabaugh, and Jeff Nelson. Garbage collection of actors. In *ECOOP/OOPSLA '90 Proceedings*, pages 126–134. ACM, August 1990.

[LKL91] J. Bryan Lewis, Lawrence Koved, and Daniel T. Ling. Dialogue structures for virtual worlds. In *Proceedings ACM SIGCHI '91*, pages 131–136, 1991. (IBM Watson R.C., Veridical User Envs.).

[McF91] Tim McFadden. *Cyberspace. first steps*, chapter Notes on the structure of cyberspace and the ballistic actors model, pages 334–362. MIT Press, 1991. Editor: Benedikt, Michael.

[Mes90] Jose Meseguer. A logical theory of concurrent objects. In *Proceedings ECOOP/OOPSLA '90*, pages 101–115, October 1990. (SRI International).

[MR89] Naftaly H. Minsky and David Rozenshtein. "controllable delegation": An exercise in law-governed systems. In *Proceedings OOPSLA '89*, pages 371–380, October 1989. (minsky@aramis.rutgers.edu).

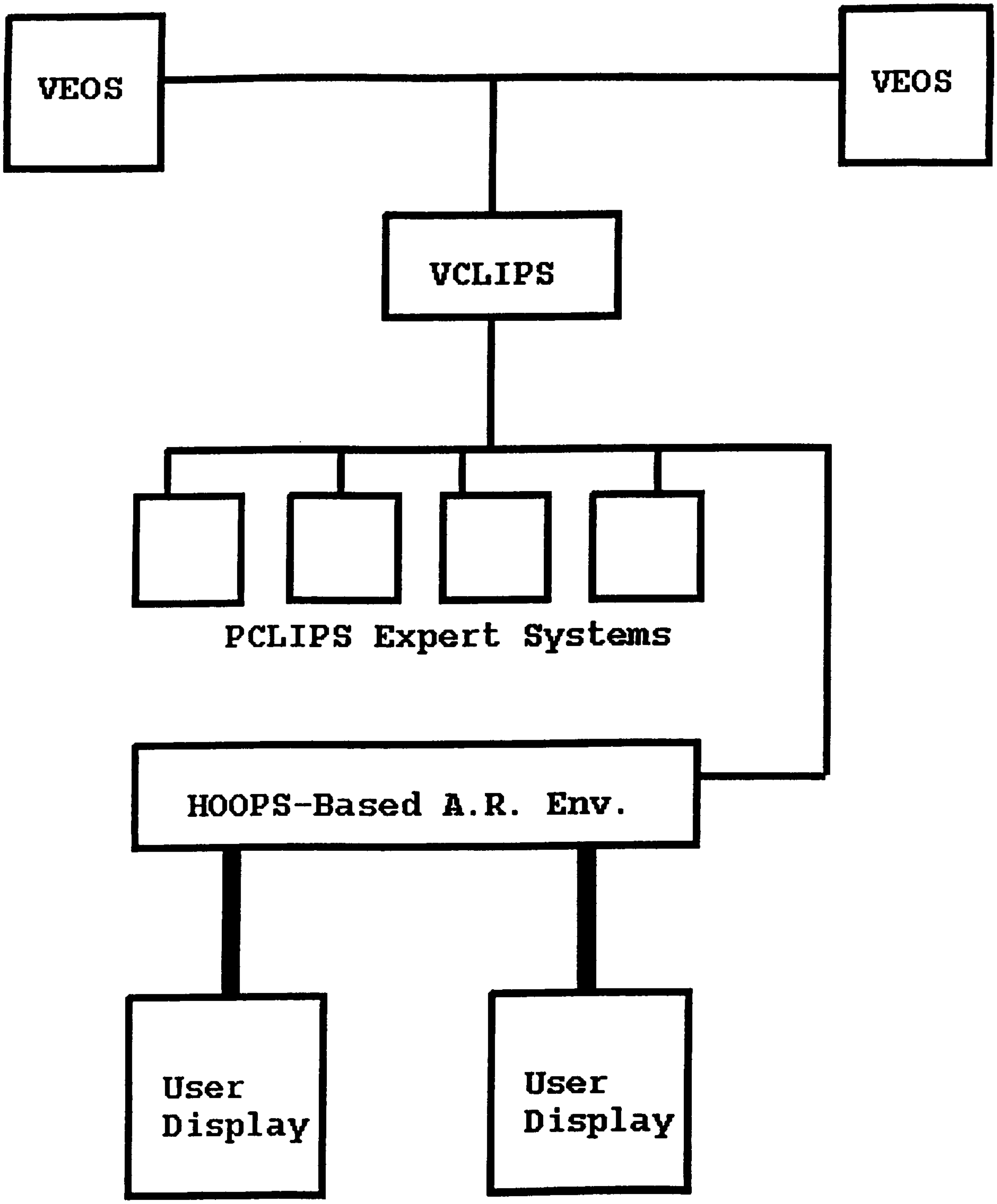
[Rub90] Kenneth S. Rubin. Reuse in software engineering: An object oriented perspective. *IEEE Publication Num. CH2343-1/90*, pages 340–346, 1990.

[SLGS92] Chris Shaw, Jiandong Liang, Mark Green, and Yunqi Sun. The decoupled simulation model for virtual reality systems. In *Proceedings ACM SIGCHI '92*, pages 321–328, 1992. (Univ. of Alberta).

[WH91] Jakub Wejchert and David Haumann. Animation aerodynamics. *Computer Graphics*, Vol. 25, No. 4:19–22, July 1991. ().

[ZCW+91] Robert C. Zeleznik, D. Brookshire Conner, Matthias M. Wloka, Daniel G. Aliaga, Nathan T. Huang, Philip M. Hubbard, Brian Knep, Henry Kaufman, John F. Hughes, and Andries van Dam. An object-oriented framework for the integration of interactive animation techniques. *Computer Graphics*, Vol. 25, No. 4:105–111, July 1991.

VCLIPS Architecture



Lincoln Logs Virtual Factory of the Future