SANDIA REPORT

SAND97-2315 • UC-810
Unlimited Release
Printed September 1997

Network-Based Collaborative Research Environment LDRD Final Report

Brady R. Davies, Michael J. McDonald

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Sandia National Laboratories

SF2900Q(8-81)

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A04
Microfiche copy: A01

SAND 97-2315
Unlimited Release
Printed September 1997

Distribution
Category UC–810

Network-Based Collaborative Research Environment LDRD Final Report

Brady R. Davies
Integrated Micromechanics, Microsensors & CMOS Technology Department

Michael J. McDonald
Intelligent Systems Principles Department

Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-1080

Abstract

The Virtual Collaborative Environment (VCE) and the Distributed Collaborative Workbench (DCW) are new technologies that make it possible for diverse users to synthesize and share mechatronic, sensor, and information resources. Using these technologies, university researchers, manufacturers, design firms, and others can directly access and reconfigure systems located throughout the world. The architecture for implementing VCE and DCW has been developed based on the proposed National Information Infrastructure or Information Highway and a tool kit of Sandia-developed software. Further enhancements to the VCE and DCW technologies will facilitate access to other mechatronic resources. This report describes characteristics of VCE and DCW and also includes background information about the evolution of these technologies.

Acknowledgments

The authors thank Dan Small, Robert Anderson, and Scott Gladwell for their significant contributions to this research.

Contents

Figures

Tables

Abbreviations and Acronyms

AGV	Automated Guided Vehicle
AMT	Advanced Manufacturing Technologies
ARPA	Advanced Research Projects Agency
ASRS	Automated Storage and Retrieval Systems
ATM	Asynchronous Transfer Mode
CAD	Computer-aided design
CAE	Computer-aided engineering
CMU	Carnegie Mellon University
CPU	Central processing unit
CNC	Computer Numerical Control
CRADA	Cooperative Research and Development Agreement
DAVE	Digital Audio Video Environment
DCW	Distributed Collaborative Workbench
Deneb	Deneb Robotics, Inc.
DH	Denavit Hartenberg
DIS	Distributed Interactive Simulation
DOD	Department of Defense
DOE	Department of Energy
DSI	Defense Simulation Internet
FoF	Factory of the Future
FTS	Federal Telecommunications System
FY	Fiscal year
GENISAS	GENeral Interface for Supervisor and Subsystems
GISC	Generic Intelligent System Controller
GUI	Graphical user interface
ICE	Interactive Collaborative Environment
IP	Internet Protocol
ISDN	Integrated Service Digital Network
JPEG	Joint Photographic Experts Group

JPL	Jet Propulsion Laboratory
LAN	Local area network
LANL	Los Alamos National Laboratories
LDRD	Laboratory-Directed Research and Development
LLNL	Lawrence Livermore National Laboratories
MfgE	Manufacturing engineer
NASA	National Aeronautics and Space Administration
NII	National Information Infrastructure
NTSC	National Television Standards Committee
OLP	Off-Line Programming
ORNL	Oak Ridge National Laboratories
PNL	Pacific Northwest Laboratories
PSVC	Packet Switching Video Conferencing
RIA	Robot Industry Association
RIPL	Robot Independent Programming Language
RTDP	Robotic Technology Development Program
Sandia	Sandia National Laboratories
SGI	Silicon Graphics
SMART	Sequential Modular Architecture for Robotics and Teleoperation
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VCE	Virtual Collaborative Environment
VME	Virtual Module Europa
VR	Virtual reality
WAN	Wide area network

Intentionally Left Blank

Network-Based Collaborative Research Environment LDRD Final Report

Introduction

As the technologies for sharing information assets continue to improve and proliferate, the importance of sharing other kinds of assets has become more apparent. In an environment of competing resources and capital, the ability to readily and reliably use assets operated by other industries, universities, research labs, and government agencies may prove to be a crucial advantage. Timely and efficient collaboration between entities has become increasingly important as monetary resources of entire industries expand or contract in response to rapid changes in demand for products, dissolution of political barriers, and adoption of stringent environmental and commercial legislation. The Virtual Collaborative Environment (VCE) and Distributed Collaborative Workbench (DCW) technologies described in this report provide the environment where flexible and efficient integration, interaction, and information exchange between disparate entities can occur.

The schedules and work procedures of many disciplines are being revolutionized as the tools and infrastructure for information sharing and task collaboration between entities separated by significant geographic distances are developed and enhanced. Technology is available that could enable most white-collar professionals to efficiently telecommute for at least a portion of their work week. In many organizations, corporate structure and policy are the only barriers to wide-scale employee telecommuting. This evolving computer network technology can revolutionize the ways that human resources and hardware resources are deployed and accessed.

Sandia National Laboratories (Sandia) has developed an architecture and system that supports the remote integration and sharing of hardware resources. This architecture is based on the proposed National Information Infrastructure (NII) or Information Highway and the Sandia developed Generic Intelligent System Controller (GISC) library of software modules.[1] The architecture facilitates the synthesis, programming, integration, control, and operation of multiple, intelligent programmable machines (such as robots and machine tools), input devices (such as computer mice, force/torque-reflective feedback devices, and sensors), software tools (such as simulation code, control algorithms, and graphical models), and sensors. Thus, with an appropriate set of these various software and hardware devices, computer-controlled resources that are owned and operated by different geographic and legal entities can be individually or simultaneously programmed and controlled from one or more remote locations.

Sandia also built and tested a prototype architecture and system that proved the feasibility of the VCE and DCW technologies. Programming and control of several robots located at Sandia in Albuquerque, New Mexico, was demonstrated from such remote locations as Washington D.C., Washington State, Detroit, and Southern California.

Remote programming and control of intelligent machines will create significant opportunities for the sharing of expensive capital equipment. Disparate electromechanical resources can be shared in a manner similar to the way supercomputers are accessed by multiple users. Using the VCE and DCW technologies, it will be possible for researchers to perform hardware validation of models and algorithms by remotely "borrowing" appropriate hardware resources. Manufacturers will be able to model, simulate, and measure the performance of prospective robots before selecting suitable robot hardware for an application. Designers will be able to access Computer Numerical Control (CNC) machining centers across the country to fabricate prototype parts when validating their product designs.

Benefits of Using VCE and DCW Technologies

Economic competitiveness in the future will depend on aggressive technology development directed toward short product-to-market cycles. Increasingly, teaming between business entities will be adopted as a key business strategy to reduce the cost and time of technology development. VCE and DCW technologies accelerate this development by taking advantage of high-speed network technology and by coupling information directly to the operation of integrated systems of machines.

While it is practical and almost instantaneous to transmit information and computer data across long distances, it is impractical and very slow to move equipment and other hardware even short distances. During the processes of developing new technologies, designing new products, and even testing new theories, the tasks of testing, prototyping, and evaluating must be performed on actual hardware and machines because models of hardware almost never behave exactly as the real hardware. VCE and DCW technologies provide the capability of configuring and accessing distributed hardware and computer resources to validate new technologies and processes, test new theories, evaluate designs and processes, and even prototype new products without requiring that the hardware components be located near the user or that new prototype hardware be fabricated.

Not only do VCE and DCW technologies provide convenient, efficient, and cost-effective access to mechatronic resources, they also provide the ability to rapidly synthesize data and information in new and useful ways. The breadth of potential applications for these technologies ranges from algorithm testing by researchers to system prototyping by systems integrators to design validation by manufacturers. New applications will be spawned as the accessibility and the capabilities of the VCE and DCW technologies continue to improve.

VCE and DCW technologies also provide a seamless interchange between virtual and actual devices. For example, if a robot module is selected as virtual, then a simulated robot (typically a software routine running on a Virtual-Module-Europa

(VME) system) is automatically connected to the system. If the model is correct, then this virtual robot reacts to the control commands exactly as though the real robot were connected. If the real robot is selected, then the system automatically reconfigures itself so that the actual robot is being controlled. This interchange is possible because the subsystem interfaces for both the virtual and actual robot are identical. Thus the entire integration (from operator interface to path planners to sensors) can be configured and tested before the real robot is procured. Once the robot is procured, then the same software that was used to drive the simulation is used to drive the actual device. This provides inherent accuracy in simulations and the ability to seamlessly shift from virtual devices to actual ones.

Using VCE and DCW technologies frees system designers and module developers from typical concerns. The system designer needs to consider only the functional characteristics of the system modules; details pertaining to module algorithms and processes are not relevant to the system designer's implementation planning. Similarly, the module developer needs to consider only the internal workings of the module and its external interfaces; application-specific details are not relevant. The development environment ensures that the library of modules can be continually expanded and reconfigured as necessary.

Technology Description

This section presents general definitions and characteristics of VCEs and the DCW, including how these technologies use high-speed computer networks.

Virtual Collaborative Environments (VCEs)

A VCE is the technology used to remotely share mechatronic devices among a group of participants[2]. Mechatronic devices are computer-controlled, electromechanical devices. Examples of mechatronic devices include robots, Automated Guided Vehicles (AGVs), numerically controlled machine tools, and Automated Storage and Retrieval Systems (ASRS). In theory, any computer-controlled, electromechanical device can be incorporated into a VCE, provided that the device has an electronic communication interface. Examples of potential VCE-compatible devices are electron microscopes, automobile engines and subsystems, drawbridges, and railroad switches. As devices are redesigned for computer control and monitoring, the breadth of potential VCE applications will expand.

A set of VCE-interfaced devices can be accessed and shared similar to the way supercomputers are shared by multiple users at disparate locations. The VCE-interfaced devices can also be used by multiple users for different applications and in different ways, as supercomputers are used for very different applications by various users. In a multi-user supercomputer environment, users access a supercomputer from remote locations and perform various computing, analysis, simulation, and rendering tasks. The supercomputer is brought virtually to the location of the user, even though the user may be located thousands of miles away from the actual computer hardware. VCE technology can provide similar multiple-user access to mechatronic devices from remote locations.

VCE users interact with virtual mechatronic devices through graphical representations of the devices. Graphical representations of mechatronic devices are generated at the work location of the user from information transferred over a computer network. Although the user can interact with the virtual devices in different ways, the VCE environment should support a high level of interaction without unnecessarily burdening the user with operational formats, details, and parameters.

There are similarities between the VCE technology developed at Sandia and technologies for remote machine access that were developed at other institutions such as the Jet Propulsion Laboratory (JPL) and the National Aeronautics and Space Administration (NASA). JPL and NASA have developed technology to teleoperate devices located in space or other very remote environments. These types of systems have highly specialized interfaces and interconnections and rely heavily on human-in-the-loop control techniques. Sandia's DCW-configured VCE system incorporates generic interfaces, system reconfigurability, graphical programming, and automatic programming and operation.

Distributed Collaborative Workbench (DCW)

The DCW is a set of system integration technologies that use a library of software tools to provide modular reconfigurability of multiple devices.[3] The devices that can comprise a system include the same kind of devices that can be accessed in the VCE environment—namely, electromechanical devices with computer interfaces (mechatronic devices), sensors, software tools, and input devices. The architecture promotes an environment where modules associated with various hardware and software devices can be readily reconfigured and modified. This environment provides for generic interconnectability, inherent control system stability, and rapid reconfigurability of multiple-device systems. The environment also has a testbed for developing and applying system integration technology in the form of a software tool kit. The tool kit describes capabilities and provides configuration and assembly of agents. Agents (for example, mechatronic devices) support either software and/or hardware functionality through a communication interface.

The DCW uses an information architecture approach to systems integration that allows merging of the software environment with the hardware environment. This environment is inherently "plug-and-play" with a structure that enables software developers to easily insert new models, modules, and techniques without extensively reworking existing software. Agents are integrated through communication interfaces. Software drivers translate the generic commands into the specific instructions required by each software agent. By defining the driver interface, virtual models of the device communicate in the same manner as the actual hardware, thus creating a seamless interchange between virtual and actual hardware. That is, the same communication code drives the virtual hardware and the real hardware.

High-Speed Computer Networks

Various local area networks (LANs) and wide area networks (WANs) networks have been assembled throughout the world. The largest collection of such computer networks is called the Internet. The Internet is a prototype of the universally accessible, high-speed networks that eventually may form a national information superhighway or National Information Infrastructure (NII).

As currently configured at Sandia, the VCE and DCW technologies require high-speed (greater than one million bits per second) network transmission to accommodate the high-volume data communication requirements of near real-time video. Dedicated-high speed networks are used to obtain the necessary performance. These dedicated high-speed networks (usually T1 networks) use Internet protocols but allow significantly higher bandwidth data transmission than is possible over the normal Internet infrastructure. While such high-speed networks are not yet universally accessible, various initiatives are being formulated to develop the infrastructure and associated technologies necessary to create them.[4,5] Additionally, continuous research is being conducted to provide tools and technologies that might significantly reduce the amount of data required to use the VCE and DCW technologies effectively.

VCE AND DCW Components

The architecture embraced by the VCE and DCW technologies is based upon reliable and timely communication of data and control information. Historically, adequate communication between computers has been very problematic, particularly when computers of different manufacturers are employed. Several years ago, Sandia began developing a library of communications and controls software that could be readily configured to accommodate diverse computer communications and controls situations.

An important characteristic of the VCE and DCW technologies is the ability to add new modules without having to reconfigure the entire system. This simplifies the incorporation, testing, and enhancement of new technology approaches by remote users. A remote experimenter provides only the increment of technology necessary for testing a new concept or approach, which reduces the time and cost of technology development while stimulating cross-institutional teaming. An innovative advance improves the entire system and each participant's associated technology.

Information Architecture

The DCW technology is based upon a development philosophy known as the Generic Intelligent System Controller (GISC). GISC is an agent-based client-server approach to systems integration. It consists of a library of software modules called GISC Kit that has been developed and applied in several robot system integration projects.[6] Agents are classified as either supervisors or subsystems. Supervisors, which are software resources, orchestrate the activities of intelligent subsystems to perform tasks using either software or hardware. Various supervisor and subsystem agents have been integrated into the DCW tool kit. These tools have been developed by the national labs, universities, and industry; and the library of tools is continuously expanding. In the DCW environment, the GISC approach treats each subsystem agent as being intelligent such that generic commands or information can be passed to the subsystem agent and executed without further monitoring by the supervisor agent. Agents are connected through a communication link using either UNIX sockets or the Sandia-developed GENeral Interface for Supervisor and Subsystems (GENISAS) connections. GENISAS is a GISC Kit module that facilitates communications between computing hardware.[7]

GISC coordinates and integrates the operation of diverse subsystems to accomplish complex information-processing tasks. GISC is used to combine machines with sensors and computer models, resulting in integrated mechatronic systems that can automatically execute remote system operations.

GISC is an intelligent control-system approach that uses information, in the form of computer models, to enable automation of system programming. All robot operations (both computer-planned and operator-programmed) are verified with computer models to validate safe operation. Intelligent decisionmaking algorithms evaluate the known environment represented by a world model, plan safe robot motions, and then automatically generate the machine control sequences necessary to execute the desired robot motions. The use of computer models for error detection and recovery prevents unsafe actions.

Supervisor Agents

Supervisor agents control the subsystem agents using time-based or event-based control methods. An example of a time-based control element is a joy stick. Using a joy stick as an input device, the user interacts with the subsystem in real-time by entering tasks for the subsystem to perform. The subsystem responds immediately to the user's commands. An example of an event-based control element is a task planner. A task planner specifies the sequence of operations required to achieve a given task. In this case, the complete sequence of events is described before any operations are initiated, and the completion of prior events initiates subsequent events. Either a time-based or control-based supervisor can be plugged into the system simulation as needs dictate.

Subsystem Agents

Subsystem agents receive and execute commands from supervisor agents. Subsystem agents are created from classes of components like robots, manufacturing equipment, intelligent end-effectors and vision systems. Each subsystem resource is completely characterized by three pieces of information: hardware type, behavior mode, and capabilities list. Hardware type includes information such as device model and type, as well as computing resources or sensors associated with the device. Behavior mode describes the operational behavior of the agent. A subsystem agent such as a robot could be configured with several behavior modes, ranging from an autonomous mode under sensor control, to a teleoperation mode with various input control devices such as a force-reflecting master, or even to a programmable mode using high-level decision planners. The capabilities list specifies the type of operations that the particular system can perform. Items in this list typically contain commands such as "get tool," move home," and "move along path." New elements are added to the capabilities list as new agent capabilities are developed.

Software drivers residing within a subsystem agent intercept commands and requests for action, translate these commands into agent-specific commands and data, and then translate the results into generic commands and data that are understood by the rest of the virtual environment. Using this interface-driven approach, agents become a collection of modules that can be assembled over a network to form integrated systems from virtual and/or actual devices. This approach defines a grammar for interfaces that allows continual expansion and revision. Thus in place of a standard, there is a defined process that provides an environment for adaptation and modification.

Subsystem-Behavior Software

The Sequential Modular Architecture for Robotics and Teleoperation (SMART)[8] and the Robot Independent Programming Language (RIPL)[9] are GISC Kit modules used to describe the behavior modes of subsystem agents. SMART provides a real-time control architecture for constructing telerobotic systems consisting of input devices, robots, sensors, and output devices. SMART modules can run synchronously and can be distributed arbitrarily across multiple Central Processing Units (CPUs) connected across networks. A robot server, created with the RIPL, is connected via network to a supervisor resource, thus completing the communications path from the supervisor to SMART. Using these software packages, it is possible to build and reconfigure flexible robot servers that incorporate different input devices and sensors operating over networks.

VCE AND DCW Technology Attributes

This section highlights several attributes of the new technologies that specifically address the needs of diverse users. These attributes include plug compatibility and access to software and hardware resources that may be used for a variety of purposes such as research, testing, evaluation, collaboration with partners, and training.

Plug Compatibility

While many of today's automation tasks can be performed using available robotics technologies, future automation tasks will probably require significantly new developments in technology and information access. As robotics technology rapidly advances, the enormous amount of accompanying information makes it almost impossible for robotic users to keep up-to-date with new developments in the field. New products are arriving on the market faster than most users can evaluate them. It is virtually impossible for robot users to stay abreast of current technology while simultaneously attempting to implement other new technology into robot systems and leveraging these new technologies to develop even better capabilities. Only a plug-compatible, modular, reconfigurable architecture (such as the DCW environment) provides an efficient means for evaluating, testing, and using new products and technologies.

Defining interfaces between various components makes it possible to achieve this "plug-and-play" environment for integrating systems. Rather than attempt to dictate a standard interface, a preferred approach is to define a grammar from which the actual interface will evolve as the market dictates. The DCW technology offers such a grammar, resulting in lower development costs while ensuring that the system works correctly the first time.

Tool Kit Accessibility

Not only does this integration tool kit need to be modular, reliable and expandable, it also must be readily accessible by all members of the systems integration team—a team that may be dispersed over wide geographic regions. For example, Sandia engages in system integration activities with team members located at other national laboratories ranging geographically from Tennessee to the state of Washington. Additionally, Sandia interacts with a variety of geographically dispersed industrial partners, providing these partners with various system components and integration expertise. Sandia is also involved in coordinating the robotic research efforts of several universities and providing research resources to them. During the summer, a number of faculty and student researchers often visit Sandia for extended periods of time to use Sandia's resources.

Interactions with industry, other national labs, and universities are often difficult and complex, but these interactions are beneficial to all parties. Leveraging the resulting technology is well-worth the complexity of the interactions. DCW technology will simplify these interactions by making the integration and data exchange processes easier and providing remote access to common resources.

Virtual Hardware Evaluation

The hardware aspects of production, manufacturing, or processing systems is very expensive. For example, robot systems that are being designed to remediate hazardous waste sites within the Department of Energy (DOE) are projected to cost hundreds of millions of dollars. Evaluating these systems before committing significant expenditures of capital is important to ensure that the system will work as designed. It is inherently expensive and difficult to modify hardware once the fabrication process has been initiated. If system flaws can be identified in the design phase of a project, a significant amount of time and money can be saved during the implementation and operation phases. The VCE and DCW technologies will provide an efficient means for verifying and validating designs before the initiation of fabrication.

Hardware Resource Access

Several testbed systems for evaluating new concepts for robot control and operation are being designed and procured for the waste cleanup initiative. It is important to make these various testbed systems accessible to the various collaborators. These collaborators are dispersed geographically from each other and most are located remotely from the testbed system. Requiring individual researchers to travel to testbed locations to perform their research is expensive and inefficient. Not only is it important to make these test beds available to researchers, but it is also important to make the test beds available to operators for training. Training is time-consuming; and it is highly desirable to provide cost-effective and realistic training using simulators before the actual hardware becomes available. The VCE was developed to facilitate remote access to hardware resources like these test beds for research and training purposes.

SANDIA's DCW-Configured VCE System

A prototype VCE was integrated and tested using the DCW technology developed at Sandia. The prototype system provides connectivity between remote computer workstations, local computer workstations, robot controllers, video systems, video controllers, robot subsystem controllers, and various graphics and video display systems. The system was configured using information obtained from the Information Models Report in Appendix A and an evaluation by potential users of the system described in Appendix B.

Figure 1 presents an overview of the DCW-configured VCE assembled at Sandia. Local networks are connected via high-speed communications links (ethernet). Network connections between the remote user network and Sandia's local network are through a dedicated communications link.

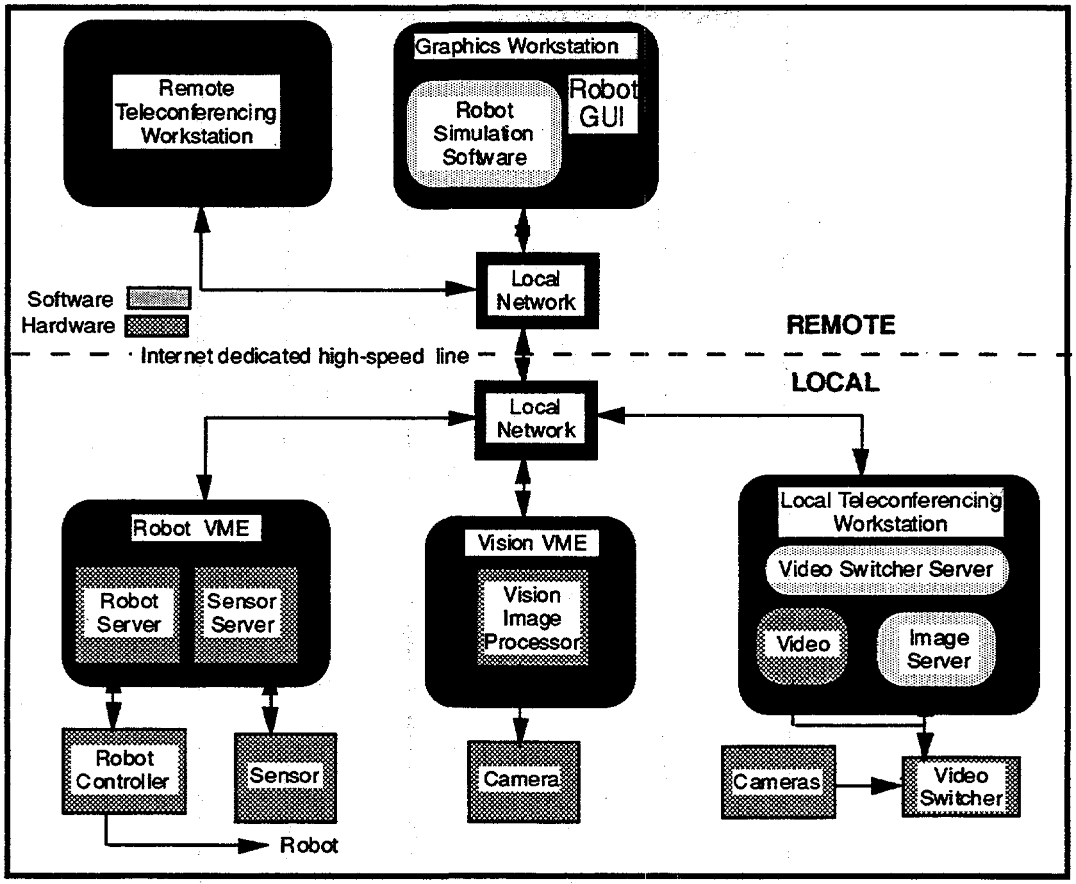

Figure 1. A DCW-Configured VCE System.

Remote VCE User Configuration

In the current configuration of Sandia's VCE, a remote user requires two computer workstations: one workstation handles all video conferencing and video interfacing while the other workstation serves as the user interface between the robot and other mechatronic devices. The user interface between the robot and other mechatronic devices is configured as a graphics-based control system. Interaction is provided to the user through a Graphical User Interface (GUI) tool that allows the user to quickly and easily program, control, monitor, test, and configure the robot and other mechatronic devices.

Video Conferencing and Video Interface

As shown in Figure 1, the Sandia remote user sends and receives video using a UNIX workstation with both audio and video hardware and teleconferencing software. The remote teleconferencing workstation is configured to handle video capture, processing (including compression), transmission, reception, processing (including decompression), and display. Teleconferencing hardware and software manufactured by both Silicon Graphics and Sun Micro Systems have been successfully used at Sandia.

Graphical Programming

A graphical representation of the robot and its environment, which corresponds closely with the real robot and its environment, is modeled and displayed using robot simulation software and a graphics workstation. The graphical representation of the robot and its environment is called the *world model.* A graphics workstation that runs the simulation and display software serves as the primary user interface for programming, commanding, and otherwise interacting with the actual robot. Simulation software is used to display the world model in real-time so that the operator can validate the robot's motion. Simulation software is also used for visually previewing the robot's motion to validate safe operation and to verify correct task performance before the actual robot moves.[10] The simulation also performs mathematical automatic-collision checking and safe-path verification. A simple, automatic path-planning algorithm has been implemented in the Sandia VCE to further enhance the user interface.

Robot position sensors (encoders or resolvers) update the world model robot configuration in real time. This information is used to drive the graphical representation of the robot in the world model so that the graphical and physical robots are continuously synchronized. Since the world model contains information about the robot's joint range of motion, joint speed limits, joint acceleration limits, and kinematics, the graphical representation of the robot workcell can be configured to warn of impending motion problems before the actual robot begins to move.

Construction of the World Model

The world model is constructed from Computer-Aided Design (CAD) modeling information and integrated sensor data. CAD models containing geometric information about the robot and its workcell are imported into the simulation software database and displayed as the world model. Generally, additional information is necessary to ensure collision-free robot motion and to allow the robot to perform tasks correctly. Information about the location of a workcell component may be inaccurate or incomplete, or information about the shape and location of a workcell component may be missing. Workcell components that have already been geometrically modeled but that are inaccurately positioned can be correctly located using visual targeting technology. Components that have not been modeled can be accurately modeled and located using a scanning laser technology called structured lighting.[1]

Local VCE Configuration

At the local VCE site, a teleconferencing system node is configured to facilitate video transfer between the remote and local sites. Video signals from stereo cameras for the visual targeting system, various workcell-mounted video cameras, and the video conferencing cameras are displayed, processed, selected, transmitted, and received by the teleconferencing system.

A vision processing unit, housed in a VME rack, has been configured to facilitate rapid video digitizing, compression, and processing for vision tasks. A second VME rack contains computing equipment for robot control and sensor-processing. The robot controller receives robot commands through the RIPL format. Communication between the robot controller, sensors, and computers has been facilitated using other GISC Kit modules.

Local Control

Servo control in a DCW-configured VCE system is performed local to the machine in an intelligent subsystem. High-speed sensing subsystems provide the necessary sensory inputs to a subsystem's motion-control algorithms to perform, for example, force-controlled interactions with the environment. Real-time sensing also allows in-process modification of machine motions to perform, for example, surface-finishing operations on machined parts. It is important that high-speed, sensor-based, servo-controlled operations be executed within the subsystem controller and not by the higher level GISC supervisor. This prevents delays in the servo control loop resulting from bottlenecks in communication and computation.

Results

A large (PaR XR100, 100-pound payload, 20' X 40' workspace) gantry robot system located at Sandia in Albuquerque, New Mexico, was used to initially develop and demonstrate the feasibility of VCE technology. Remote control sites located in Washington State, Washington D.C., California, Detroit, and the Albuquerque Convention Center were networked to the robot with dedicated network links. Control areas in conference rooms at Sandia were networked with ethernet to demonstrate the VCE concept locally. An IGRIP-based graphical-programming supervisory system from Deneb Robotics, Inc. and a Silicon Graphics' video-teleconferencing system formed the supervisory interface.

In typical user sessions, tasks were selected and defined by the remote user. An automated planning and programming system generated robot motion plans necessary for executing the task, and the user was able to accept or reject the plans. Once a plan was accepted, it was transmitted to the gantry robot at Sandia. The graphics system used position information from the actual robot to update the graphical robot's position and transmitted the position data to the remote user for use in monitoring the system. Real-time video images of the actual robot were also transmitted to the remote user via the teleconferencing system.

A variety of volunteer users were trained to operate the system. These users were government officials, managers, secretaries, engineers, and technicians. Training consisted of a 5-minute to 10-minute demonstration. After training, the users could update the world model, perform contact surveys (on 55-gallon waste drums, mock shipping containers and other workcell objects), and grind a variety of parts (including curved steel sections). The users frequently praised the system's simplicity of operation and its performance capabilities. See Appendix B for more information on feedback from potential users.

All milestones were completed on schedule as listed in Table 1.

Table 1. Milestone Status for Network-Based Collaborative Research Environment LDRD

Milestone	Date	Status
Develop information interfaces for local and remote systems	12/94	complete (Appendix A)
Develop prototype interface systems for local and remote operators	6/95	complete
Demonstrate prototype system from a remote (non-Sandia or partner) site	8/95	complete (NIST)
Publish internal report of potential user evaluations	9/95	complete (Appendix B)
Develop full-scale interface systems for local and remote operators	5/96	complete
Operate Sandia robots with full-scale system from a national conference	8/96	complete (RIA 1995)
Complete test and refinement phase with potential user interactions	8/96	complete
Publish final report	7/97	complete

VCE AND DCW Applications

In this section, two hypothetical applications of the VCE and DCW technologies are described. The companies and locations used in the examples are fictitious and included only to illustrate the concepts.

Access to Mechatronic Resources

The VCE and DCW technologies could be used to provide efficient access to computer-controlled equipment, sensors, and machines by researchers at universities, labs, and industry. Many universities and research labs generate good ideas, but they generally lack sufficient resources to prove the validity of their concepts. Access to mechatronic resources located at other universities, research labs, or industrial concerns will enhance the ability of researchers to test, validate, demonstrate, and further develop algorithms and solutions pertinent to their fields of research.

Consider the case of a university researcher in a small college in Maine who has developed a new fuzzy-logic real-time control algorithm that is applicable to manufacturing operations such as machine vision-based shaft insertion. Using a DCW-configured VCE, the researcher could test and demonstrate the algorithm by connecting to a robot system located at an aerospace facility in Southern California. The algorithm could be tested, debugged, modified, and eventually developed to the point of practical use without requiring that the researcher travel across the country to iteratively test and modify the algorithm and associated computer code. Data obtained at the Southern California aerospace facility could be directly passed to the researcher's college computer system for analysis and display during the testing and validation phase. Modifications to the control algorithm and code could be implemented and tested on the robot system using a DCW-configured VCE. Even if the robot system was fully used during the day for mechanical assembly operations, it might be available for remote access during the night.

Remote Machining

The VCE and DCW technologies could also be used by a small design company to machine a complex part. Assume that this company designed a prototype system composed of a series of large aluminum components with intricate contours. The company, located in Arizona, could use a VCE to access an appropriate 5-axis milling machine located in Seattle, Washington, and literally build the parts remotely with its own tool-control information. The designer could interact with the milling machine during the machining process to ensure that proper tolerances and fits were maintained. Completed parts could then be express-mailed to the Arizona company for integration with locally produced parts.

Future Enhancements

Developers have identified three areas for system enhancements. These areas cover the development of improved generic interfaces, the provision of more intuitive control information, and exploration of approaches for safe VCE control.

Generic interfaces

To make mechatronic devices readily accessible to a wide range of different users for varying applications, improved generic interfaces must be developed. These interfaces must 1) facilitate useful interaction between the user and various mechatronic devices accessible by the user and 2) enable the operators or owners of mechatronic devices to make the devices readily available for remote access. Types of interfaces include human-machine interfaces, security command and control systems, machine safety systems, data formats, programming formats, and communications protocols. Many of these components can be provided by the GISC software currently used at Sandia. A secure-access technology called TIE-In, also developed at Sandia, is being modified to incorporate the DCW software. TIE-In provides secure Internet Protocol (IP) gateway service.

Virtual Sensor/Force Reflection

One method of providing intuitive control information for safe and efficient task operation is to reflect external forces (or some other appropriate sensor measurand) experienced by the mechatronic device back to the remote user. For instance, if direct interaction between the mechatronic device and a relatively immovable object is necessary to complete a task (e.g., inserting a shaft into a hole), then the remote user might be more able to direct the robot if comparable force/sensor data were sensed by the remote user. Sensing measurands like forces typically require measurand transmission bandwidths higher than those available across a network like the Internet. One way to overcome this limitation, while still providing real-time feedback to the remote user, is to use information contained at the local model. Simulated forces or proximity measurements computed by the remote robot simulation system can be sent directly to the remote user at very high bandwidths. The simulated or "virtual" forces can then be presented to the remote user as force feedback in the case of a force-reflecting master, or in some other appropriate form.[11] Proximity sensing, such as that provided by whole-arm-capacitive proximity sensors,[12] can also be reflected to the remote user as virtual force data, or in another suitable form. Reflection of real-time sensor data increases safety and the speed of system operations.

Operation Safety

Two different operational configurations for safe VCE control are being considered. The first configuration employs the local operator as a robot motion filter. In this mode, all commands originating from a remote operator would be tested with a simulation system by the local operator before the actual robot would be allowed to execute these commands. Obviously, this scenario is not well-suited to teleoperation unless the remote operator creates a command set by "teleoperating" the remote model first, and then commands the real robot to execute the command set created by the teleoperation session. Unsafe commands could be automatically rejected by the system or the local operator could be given the option of manually accepting or rejecting unsafe commands. This approach allows great flexibility in remote user software, but it introduces an additional simulation delay between remote initiation of robot motion and actual execution.

The second configuration for safe VCE control is employed in the current DCW-configured VCE system at Sandia. In this mode, the local operator uses a video teleconferencing system to monitor all commands originating from a remote operator at the remote site. When potential problems are flagged, the local operator intercedes to ensure continuous safe operation of the robot. This mode also facilitates efficient teleoperation of the local robot by the remote operator. The local operator monitors the graphical simulation occurring at the remote site and also monitors the remote operator's interaction and commands. Future implementations will transmit data from the world model more efficiently and allow different simulation software to be used at each site.

Summary

VCE and DCW technologies let diverse users (including nonprogrammers) easily synthesize and share mechatronic systems. An integral information architecture transmits and uses information to automate the programming of machines. Further enhancements to the technologies will facilitate access to other mechatronic resources by an increasingly wider scope of remote users. VCE and DCW technologies can provide greater access to expensive capital equipment, and they can also facilitate rapid prototyping and shorten the product-to-market cycle.

References

1. B. R. Davies, M. J. McDonald, and R. W. Harrigan, "Virtual Collaborative Environments: Programming and Controlling Robotic Devices Remotely," in *Proceedings of the SPIE Telemanipulator and Telepresence Technologies Conference*, October 31–November 1, 1994.

2. R. W. Harrigan, M. J. McDonald, and B. R. Davies, "Remote Use of Distributed Robotics Resources to Enhance Technology Development and Insertion," in *Proceedings of the 1994 World Automation Congress*, Vol. 2, 1994, pp. 175–182.

3. F. J. Oppel and R. D. Palmquist, "A Virtual-to-Real Robotic Environment," in *Proceedings of the American Nuclear Society 6th Topical Meeting on Robotics and Remote Systems*, February 6–9, 1995.

4. "Outlook," *U.S. News and World Report*, February 27, 1995, pg. 14.

5. A. Gore, "Vice-President Proposes National Telecommunications Reform: Press Release dated January 11, 1994," The White House Office of The Vice President, 1994.

6. J. M. Griesmeyer, M. J. McDonald, R. W. Harrigan, P. L. Butler, and J. B. Rigdon, "Generic Intelligent System Control (GISC)," Sandia Internal Report, SAND92-2159, Albuquerque, NM: Sandia National Laboratories, 1992.

7. F. J. Oppel, R. D. Palmquist, and B. R. Davies, "Remote Access and Modular Integration of Mechatronic Resources Using Virtual Collaborative Environments (VCE) and the Distributed Collaborative Workbench (DCW)," ARPA Workshop white paper, March, 1995.

8. R. J. Anderson, "SMART: A Modular Architecture for Robotics and Teleoperation," in *International Symposium on Robotics and Manufacturing (ISRAM '93)*, Santa Fe, NM, April, 1993.

9. D. J. Miller, and R. C. Lennox, "An Object-Oriented Environment for Robot System Architectures," *Control Systems, IEEE*, Vol. 11, No. 2, 1991.

10. M. McDonald and R. D. Palmquist, "Graphical Programming: On-Line Robot Simulation for Telerobotic Control," in *International Robots and Vision Automation Show and Conference Proceedings*, pp. 22-59 to 22-73, 1993.

11. R. J. Anderson, and B. R. Davies, "Using Virtual Objects to Aid Underground Storage Tank Teleoperation," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, 1994.

12. J. Novak et al., "Collision Avoidance During Teleoperation Using Whole Arm Proximity (WHAP) Sensors Coupled to a Virtual Environment," in *Proceedings of SPIE International Symposium on Optical Tools for Manufacturing and Advanced Automation*, 1993.

Appendix A: Information Models for Virtual Collaborative Environments

The first milestone of this LDRD project was to develop information interfaces for local and remote systems. The report contained in this appendix represents achievement of that milestone. Please note that this report was written in 1994 and the information reflects the status of the LDRD project at that time.

Overview

Sandia's Virtual Collaborative Environment (VCE) research goal is to produce an information and communication architecture for the collaborative sharing of mechatronic resources. The architecture will be used by a broad spectrum of Sandia and partner programs ranging from advanced manufacturing to environmental restoration. This report defines VCEs, reviews existing VCE component technologies and their applicability, describes typical VCE modes of use, outlines VCE system design archetypes, and develops an information sharing model to implement VCEs with existing robot control and networking technologies. The information architecture will lead to interface standards, command and control languages, protocols, and grammars for integrating heterogeneous subsystems into functional distributed systems.

Introduction

VCEs are a new approach for sharing hardware resources. These environments extend remote software access and intelligent system control technologies to the shared control of distributed equipment for experimentation, integrated system prototyping, and system operation. Comprehensive computer models that incorporate real-world sensing represent all system information for VCE participants. Software and hardware modules, interfaced to the VCE through intelligent software drivers, condition commands and queries to control specific subsystems and maintain model accuracy.

VCEs accelerate technology development by taking advantage of the rapidly growing National Information Infrastructure (NII) and coupling information directly to the operation of integrated systems of machines. VCEs stimulate technology development by minimizing capital costs and maximizing team-based technology development. With VCEs, technology development teams can be widely disbursed with mechatronic hardware far distant from user sites. VCEs also accelerate the use of technology because much of the controlling software can be directly complied for distributed manufacturing scenarios.

VCEs use peer-to-peer and client/server agent-based information architectures to allow sharing of both software and hardware resources for team-based technology development. This distributed technology development environment reflects the basic structure of the distributed manufacturing environments envisioned for agile manufacturing and factory of the future concepts.

VCE systems must be easy to use by application users and developers, and they must be easy to extend by researchers. System safety must be ensured through integral simulation and sharing of critical information between remote and local operators. The system must be capable of accessing robotics resources from partner sites.

A development system is being built that supports manufacturing operations including machining, assembly, and welding. The system will also support waste handling and remediation. The system will be configured in such a way that it is extensible to automatic configuration systems such as Carnegie Mellon University's (CMU's) Onika and Sandia's Distributed Collaboration Workbench (DCW).

Enabling Component Technologies

VCE development draws from many technologies. Networking and communications technologies are used to electronically tie together various work sites, and video teleconferencing systems are becoming readily available. Software sharing and collaboration technologies are used to share and distribute software across a wide potential user group. Robot control technologies are used to safely and efficiently control remote robot systems. System configuration technologies are being developed to help users rapidly design complex applications.

This research develops systematic methods to use these separate component technologies in systems. A brief overview of these technologies follows.

Wide Area Network (WAN) Technologies

New WAN technologies are becoming practical and widely available. This section discusses the applicability of these technologies for VCEs.

Dedicated point-to-point network technologies exist in a number of forms. Speeds range from 56Kb Integrated Service Digital Network (ISDN) connections through 1.5Mb T1 links to 45 Mb T3 links. These links can be used to connect two business sites (as between the Sandia locations in New Mexico and California) or as part of bigger systems.

The Internet is an example of a routed system that employs many point-to-point connections. To the user, the Internet appears as a seemingly unified network. Access to the Internet is relatively inexpensive (given the performance capabilities) for transmitting data that is not in real-time. The heavily routed implementation, however, introduces high delay and nondeterministic response times and is therefore not optimal for time-critical components of VCEs including teleconferencing and direct control.

Dial-access 56Kb and 128Kb ISDN is available in many markets at modest costs. However, ISDN implementations vary widely across service areas: availability is limited and digital compatibility is not always available. Dial-access ISDN is currently useful for point-to-point teleconferencing and on-demand linking of small sites onto larger nets.

ISDN technology is available to allow cost-effective, high-speed dial-in connection between various subnets. While retaining the advantages of direct T1, charging is based on connect time. Furthermore, ISDN opens the possibility of rapidly establishing connections to new customers who have the technology for other connections. A single node can simultaneously connect to several locations at different statically determined speeds with the total node speed (strictly additive) being limited to 1.55Mb. When multiple applications simultaneously use the node, the combined speed cannot be more the maximum node speed.

Frame relay promises to provide a connection-oriented network that appears to the user like a routed system, while significantly reducing delay and increasing deterministic behavior. Minimal frame-relay technologies are available now for up to T1 rates. Very high-speed interfaces and true-cloud-model connectionless environments should be deployed within the next two years. Sandia's Advanced Manufacturing Technologies (AMT) net will deploy frame-relay technologies to connect Sandia and several customer sites.

Asynchronous Transfer Mode (ATM) network technology is being developed that promises very high-speed, low-latency, deterministic aspects. ATM Local Area Network (LAN) components are available now and WAN test nets are planned for experimental status on the Federal Telecommunications System (FTS) 2000 for mid-1995.

Video Teleconferencing

New high-performance teleconferencing technologies are reaching the market. Video digitizing and compression hardware is available in integrated stand-alone systems (like PictureTel), as workstation board-level products (like Parallax), and as workstation software products like Sandia's Digital Audio Video Environment (DAVE) and Silicon Graphics' (SGI's) InPerson software.

Current integrated systems use dedicated point-to-point networks and proprietary very high (100:1) video compression hardware to obtain very good transmission quality at modest network bandwidths. For example, PictureTel operates adequately on 56Kb and 128Kb data lines, and it gives very good performance at about 700Kb.

Workstation-based products typically have good (20:1 Joint Phototgraphic Experts Group [JPEG] or lower) compression and are compatible with the multi-application data networks (e.g., Internet). Performance is strongly related to the networking protocols used. For example, all systems that have been tested by Sandia on WANs using Transmission Control Protocol/Internet Protocol (TCP/IP) have significantly worse performance than those systems that use User Datagram Protocol (UDP)/IP. As a result of lower compression and general-purpose network support, workstation systems require higher network bandwidths for the same video quality as integrated systems. For example, a Parallax/Uniflix system operates marginally at 256Kb, good at 700Kb or 1.5Mb, and very good at 10Mb.

Software Collaboration Technologies

Sandia's TIE-In system provides a framework and technology for sharing software outside Sandia. Consistent interface standards are being developed for this system to ease user transition from one Sandia software package to another.[1] TIE-In also provides significant security and accounting features; and it is highly leveraged in several outreach initiatives by Sandia in design and manufacturing technology.

TIE-In technology provides a framework for establishing proxy services. TIE-In users communicate with the TIE-In system on the user side and applications communicate with the TIE-In server on the provider side. The current implementation of TIE-In supports Xwindows applications by making the user Xwindows server think it is connected to a client on the TIE-In system and by making the application client think it is connected to a server on the TIE-In system. The TIE-In system then forwards or proxies the messages between the systems.

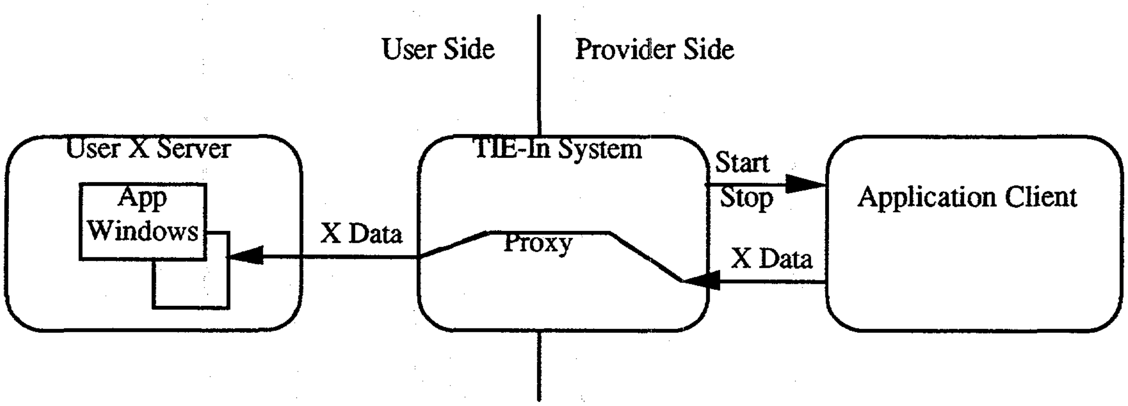

Figure A-1. TIE-In Conceptual Design.

Sandia's Interactive Collaborative Environment (ICE) is a system that allows several users to share execution of a single program. TIE-In can be designed to support this basic functionality by branching the Xwindows connections at the proxy interface to multiple servers.

[1] See http://www.sandia.gov

New communications technologies are being integrated into TIE-In. The ability to forward general UDP and TCP packets through the TIE-In proxy system has been demonstrated. The impact of teleconferencing on TIE-In performance is being evaluated and options are being explored. Initial experiments show that the existing TIE-In proxy system can support Parallax UDP-based video transmission at rates typically used for workstation/WAN teleconferencing. ICE proxy systems are being developed for use within the TIE-In framework.

Collaboration Technologies

Collaborative model-sharing technologies are being developed for multiuser virtual reality (VR) systems. Similar technologies are also being found in the multi-user Distributed Interactive Simulation (DIS).[2] MBONE[3] is being developed for the Internet to support widespread sharing of data; and it could possibly support very large numbers of people dynamically sharing the same digital VR environments. Examples of the technology exist in the VR station developed at Sandia, prototype versions of TELEGRIP developed by Deneb Robotics, Inc. (Deneb) for the ARPA Simulation Based Design program and the Department of Defense (DOD) Defense Simulation Internet (DSI)[4] system.

The basic paradigm for model sharing is to allow various simulation systems to exchange intermediate-level geometric information, such as object positions, at regular intervals or whenever the information changes. Normally, a communications technology such as broadcast UDP is chosen to eliminate synchronization delays, based on the assumption that it is better to miss information than to get stale or infrequent information. Other mechanisms can be deployed to ensure that a minimum level of information gets through. For example, in Deneb's current system, simulation systems on the network both can publish the position of a device model and its parts and can subscribe to what other simulation systems publish. In effect, the subscribing device models become slaved to the publishing masters. Published packets contain the device name, base position (transformation matrix), and joint values.

Robot Control Technologies

The Generic Intelligent System Control (GISC) approach was developed for the Robotics Technology Development Program sponsored by the U.S. Department of Energy's Office of Technology Development. GISC is an information-based strategy for integrating heterogeneous subsystems based on the concepts of modularity and extensibility. GISC provides an integration approach for designing robotic systems. GISC is an information-driven approach that leverages NII high-speed multiplexed communications.

---

[2] See ftp://taurus.cs.nps.navy.mil/pub/npsnet/mcast/FAQ_DIS, IEEE 1278-1993 & IEEE 1278.n

[3] See ftp://taurus.cs.nps.navy.mil/pub/npsnet/mcast/mbone_faq

[4] See http://www.tiig.ist.ucf.edu/

Graphical Programming is a telerobotic control technology in which operators interact with graphical simulations to test and develop robot plans. The operators can then immediately download, execute, and monitor operations that are safe and effective. Originally developed for remotely controlling robots working in hazardous environments, Graphical Programming is an appropriate tool for remote robot control.

The Generalized Interface for Supervisors and Subsystems (GENISAS) package is a programming tool for producing intelligent system command and control system client and server modules. Both GISC and Graphical Programming are being transferred to industry through Cooperative Research and Development Agreements (CRADAs) and other types of licensing agreements. Commercial replacements for GENISAS components are being sought.

The National Aeronautics and Space Administration's (NASA's) Jet Propulsion Laboratory (JPL) has developed a telerobotics system similar to Graphical Programming and has used it to remotely control robots in well-publicized demonstrations. JPL has incorporated video interface technologies in their system for rapid world-model building and better visualization. Deneb is incorporating these JPL technologies into their TELIGRIP package.

CMU has developed Onika as an iconic-based interface for programming robots.[5] Icon-based program modules developed in Onika are easily shared between users and are highly modular and reusable on a variety of robot systems. Sandia is developing the Distributed Collaborative Workbench (DCW). The DCW will facilitate rapid integration of systems constructed from known component archetypes.

Uses of Component Technologies

The previously described component technologies can significantly aid in extending the VCE concept and implementing VCE systems. This section describes how these component technologies can be employed in VCE systems and identifies the possible implications of such usage.

WAN Technologies

VCEs must operate on both existing and emerging WAN network technologies. Because the application interfaces to these technologies are very similar, converting VCE software systems from one WAN network type to another requires minimal effort. (Note that delivering the networks is a big task that is outside the scope of this work.) The key

---

[5] M. W. Gertz and P. K. Khosla, "Onika: A Multilevel Human-Machine Interface for Real-Time Sensor-Based Robotics Systems," in Proc. Of SPACE 94: The 4th Int'l Conf. And Expo. On Engineering, Construction, Feb 26-March 3, 1994, Albuquerque, New Mexico. (9 pages).

VCE work is to understand the technical advantages and limitations of the underlying WAN technologies and to tune the applications to maximize their utilization.

Teleconferencing performance, for example, is very dependent on network bandwidth and latency. Optimizing teleconferencing software for new WANs involves choosing the appropriate messaging protocol (e.g., TCP or UDP) and optimizing variables (e.g., packet size and video rates). On-demand networks might require conditioning the communications link for bandwidth or other changes so that applications do not fail in the transition process (e.g., opening the ISDN connection or ramping up the speed on an ATM link).

Connection-oriented nets (i.e., nets with nodes that are not always connected) may have a profound impact on systems where many users share the resource. This is especially true when the users have applications with dynamically changing speed needs. For example, when an ISDN user dedicates a fixed portion of the link to specific connections, the portion is unavailable until the link is broken, and thus applications that automatically adjust network load cannot take advantage of network lulls. As a result, users must coordinate and schedule use of these networks.

Tests are planned for Internet, point-to-point T1 links, and the new emerging technologies of ISDN, frame relay, and ATM WAN. The technological impact of the newest WAN technologies is difficult to predict because these technologies are currently not available or they are not fully specified.

Video Teleconferencing

Teleconferencing and video-broadcasting technologies are being used to distribute audio, video, and other visual information (e.g., white boards).

Sun/Parallax hardware is being used for its good (20:1) video compression. Other hardware might be used to overcome the marginal audio capabilities of the Sun computers that host the Parallax hardware.

Packet Switching Video Conferencing (PSVC) software is being used for teleconferencing but its TCP/IP-based communications is the major limiting factor. Uniflix is being used for broadcast systems because of its superior performance and UDP/IP-based communications.

PictureTel point-to-point teleconferencing technologies may be evaluated for use in a piggyback fashion (e.g., use Internet for data and PictureTel for audio-visual). Hardware costs for PictureTel are decreasing significantly as they enter the small-office market.

Collaboration Technologies

In their current form, TIE-In and ICE technologies are appropriate for allowing sharing of some program components. ICE technologies can be used to distribute program control information. ICE's capability to constrain program control to a single user would allow adherence to single point-of-control requirements. Supervisory programs may need to be redesigned to allow some portions of the interface to be ICE-accessible while other portions of the program are restricted.

Some visualization portions of VCE systems have graphical performance requirements that cannot be distributed over a network at the geometric primitive level. For example, graphic display rates of 100K to 1M polygons per second on 1280 X 1024 pixel 24-bit displays are common. Distributing that information at the geometric or pixel level would require Gb networks. Thus, TIE-In's geometry-level distribution technologies cannot be used for these portions of a VCE system.

Ways to use the new TIE-In technologies for forwarding application data are being developed. This forwarding capability is directly compatible with all GISC-kit network communications approaches. These technologies will allow both provider and customer sites to better secure valuable resources and data.

Robot Control Technologies

Technologies for robot command and control that were developed in earlier work by Sandia and JPL can be used for VCEs. However, multiple operators with different tasks are required in VCE systems, and VCEs are designed for resource sharing as well as for resource control. This means that new ways of using these technologies must be developed, and technologies for sharing and transforming information in real-time must be added to these systems.

While current GISC-based Graphical Programming is an appropriate VCE interface component, it does not currently allow adequate information-driven sharing of control information. Very general interfaces will be developed to maximize the capability of supervisory programs to operate and share many resources.

Component robot-control technologies being developed outside Sandia have practical implications and uses for VCE. The Sandia effort will monitor and incorporate technologies that maximize information-driven sharing of control information. Components and concepts from the JPL and CMU technologies will be used in VCE subsystems developed in this proposal, and VCE systems may be used to allow development for this or similar work.

VCE System Archetypes

When we say that any actual thing in the world (as opposed to a concept that is already abstracted) is quite simple and needs only one sort of explanation, we are, almost unavoidably, saying that it is something fairly trivial. Spoons are a great deal easier to explain than laws or trees or earthquakes or passions or symphonies, and even spoons have several aspects – culinary, metallurgical, aesthetic and what not. Anything more important than spoons is bound to have many more. Important things are, by definition, ones that have many connections and many aspects.[6]

VCE system design is not trivial. One can envision many valid VCE systems, as there are many ways to allow remote users to share mechatronic resources. The implications of this type of work are not fully understood. By developing working prototypes and building on technologies that work, we seek to further this knowledge.

This work focuses on a particular type of VCE system that features

- automatically error-checked communications for command and control protocols that ensure that commands are received, understood, and completed
- rapid, unchecked communications for sharing temporally sensitive information between systems
- automated planning software to allow users to work at an abstract task level
- simulation systems to pre-check all mechatronic system behavior for safety and efficacy.

General Model-Based VCE System Design

Sandia's VCE systems have both a remote operator who defines and controls the machine system and a local support operator who performs safety and nonautomated support operations (see Figure A-2). Additional collaborators may also share in the information analysis task and propose operational options. In some dynamic VCEs, these additional collaborators may become operators after another operator consciously relinquishes control.

[6] Mary Midgley, **The Ethical Primate**, 1994.

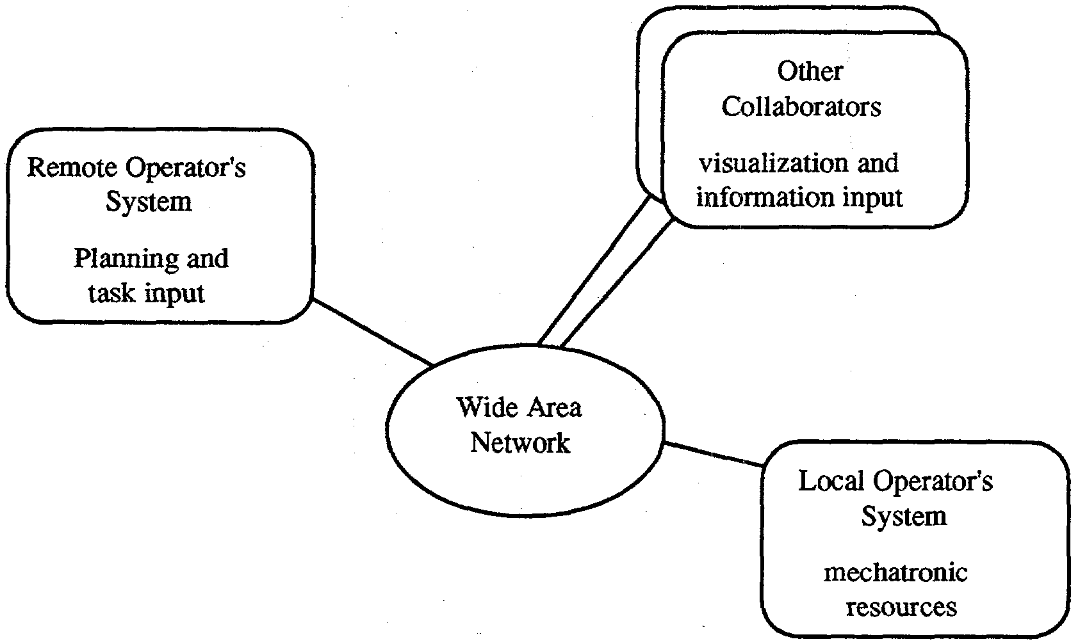

Figure A-2. General VCE System.

Computers support the jobs of all users. A network connects the computer systems to allow automatic transmission of commanded tasks and automatic sharing of information. The shared information includes geometric, property, state, task, sensed, and person-to-person data.

The remote operator's main job is to define tasks for the mechatronic system. The local operator's main job is to provide physical system support, operational safety, and operational advice. Both operators must share planning and control information to perform their separate jobs effectively.

The local operator is the main robot operator for safety and accountability purposes.[7] While both operators share responsibility for the overall safe operation of the system, the system should support the local operator's safety function by providing predictive models of the overall system and adequate veto control for operators. Providing appropriate safety tools and authority to the local operator better enables the remote operator to focus on tasks.

The remote operator's interface to the system uses a graphic-rich modeling system for task planning. Typical remote user interactions are at a task or operation level and not at a servo level.[8] Other remote users employ the same or different graphic-rich environments to monitor task planning and execution and provide advice to the remote operator.

[7] See Robot Industry Association (RIA) and enterprise operational safety standards.

[8] An independent project is ongoing at Sandia to provide remote operators with servo-level interfaces to the VCE systems. This work may support user interface options to select these new modes of operation.

The remote system generates plans and queries a safety-checking simulation system to test them for safety and validity. Results of the simulation are made available in graphical form to the local operator who might add unmodeled constraints to decide whether to allow an operation to be performed. Simulation results are also provided to the local operator who may add additional unmodeled constraints. Approved, acceptable operations are then downloaded to the real-time system for execution on the hardware.

Due to the nature of the networked environment, the safety checking simulation system can reside at any point on the network. Furthermore, multiple simulation systems can be used to distribute the separate tasks of various users. For production systems, placing the safety-simulation system at the control of the local operator can improve reliability and better enable the local operator to manage safety functions. For development systems, the remote operator may wish to be in control of the safety system and thereby maximize development flexibility at the cost of increased risk.

Direct servoing of the robots is performed on real-time computers at the local site. Use of local real-time computers eliminates nondeterministic delays from communications networks and from operating system tasks. While Sandia mainly uses VxWorks-based Virtual Module Europa (VME) systems for real-time systems development, commercial DOS-based systems are also used for the direct servoing portion of the system.

User interfaces as well as data storage and representation systems operate on non-real-time computers or on personal workstations. Currently, Unix computers provide the best libraries of highly advanced visualization and communications interface development tools, while personal computer workstations (like Macintosh or DOS) are advancing in their capabilities toward supporting these needs.

The basic model-based VCE design approaches that we have selected for further research can be called the Local Operator as Filter, the Local Operator as Observer, and the Distributed Collaborative architectures. While these approaches use similar components, they differ in how they handle the information of interaction, which influences differences in their behavior.

Local Operator as Observer Architecture

Illustrated in Figure A-3, the Local Operator as Observer architecture depends on the remote user's simulation system to filter motion requests through a simulation system that is controlled by the remote user. The first VCE system that Sandia developed used the Local Operator as Observer architecture.

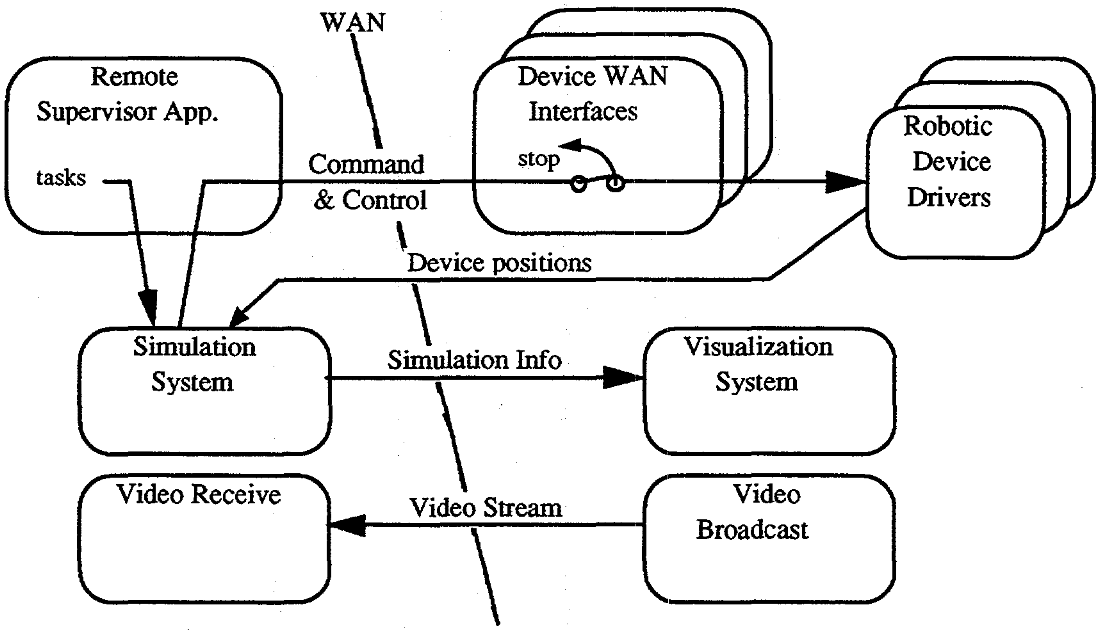

Figure A-3. Local Operator as Observer Architecture.

Brief Description

The remote operator uses a remote Supervisor to command tasks for the robots, to automatically plan, simulate, and review the robots' behavior, and to download and monitor acceptable task plans. (Note that Graphical Programming has been used for a significant part of this software.) The remote operator is responsible for maintaining the accuracy of the world model and the simulation environment, for properly using and maintaining the supervisory program, and for monitoring simulations to ensure that unmodeled constraints are adequately managed.

The local operator uses a visualization system to monitor the remote operator. The remote operator's plans are transmitted to this visualization system from the remote Supervisor. (Teleconferencing or multiuser sharing technologies can be used to transmit visualization data. If the local operator sees that the remote operator has not approved an unsafe plan or observes the robot behaving differently than simulated, then the local operator can either ask the remote operator to change the plan or interrupt the task and reject it from the system.

In Sandia's first VCE system, for example, the remote operator used a Graphical Programming system that was enhanced with workstation-based video teleconferencing. the remote operator's visual interface was converted into National Television Standards Committee (NTSC) video and transmitted to the local operator as compressed video. The local operator was in charge of the emergency stop and could electronically lock out remote operators.

Features

The Local Operator as Observer architecture provides direct access from remote supervisory programs to the mechatronic system. The operation-level system interface can be hidden from the remote operator through a sophisticated graphical programming system or provided to the remote operator for algorithm development. The local operator can interrupt or stop tasks but cannot modify them. There is little delay between commanding a task and having it executed, as it is assumed that all commanded tasks have been appropriately simulated or otherwise tested.

The remote operator builds the model, chooses how to simulate the tasks, and otherwise commands all aspects of the system. All components are under the control of the remote operator and can be changed without consulting the equipment provider.

Advantages

The Local Operator as Observer architecture provides the remote operator with the highest degree possible of unimpeded control. This architecture is very appropriate for remote developers. It provides sufficient access to allow implementation of any algorithm that can be developed locally. It is also a logical extension of traditional development techniques and provides a close analogy to developing software on the robot.

Disadvantages

The Local Operator as Observer architecture places most of the burden of safe robot control on the remote operator. This distribution of authority can overburden the remote operator with requirements that are hard to fulfill. Some remote operators seldom or never physically visit the hardware site and therefore are unaware of the special unmodeled constraints of the site. For example, robot cabling harnesses might be poorly modeled and require special care for some motions.

These architectures can disempower the local operator from providing appropriate advice and input even though the local operator might be the most qualified person for providing the advice. It is difficult for the local operator to adequately determine what the remote operator is doing. This is compounded by the fact that all pacing is done by the remote operator. Critical safety information is easily lost whenever the local operator looks away.

Placing too much authority on the remote operator gives the local operator a false sense of security. It is not likely that a local operator who is placed in a subordinate role will take control at appropriate times. This problem is analogous to that faced in the 1980s during the development of expert systems. Those efforts showed that when people were assigned a minor role in problem solving, they would not take control when the computer reasoned incorrectly. Thus, unsafe motions are more likely to be approved.

By allowing the remote operator to modify the supervisory code, the local operator may not have adequate control over program quality. It would be difficult for a local operator to know if the remote system had been programmed incorrectly, especially in the area of safety planning.

This architecture places the most expensive software components (i.e., the modeling and simulation system, planning system, and graphical monitoring system) at the remote operator's station. This distribution of ownership increases the fixed cost of establishing a remote site beyond the financial and technical ability of some users. In effect, it could force some hardware providers to use inexpensive supervisory systems, that are both inadequate and risky, to keep their customers.

Local Operator as Filter Architecture

Shown in Figure A-4, the Local Operator as Filter architecture filters all hardware motion requests through a simulation system that is controlled by the local operator. The validated simulation system simulates the effect of each motion request and advises the local operator of known safety implications.

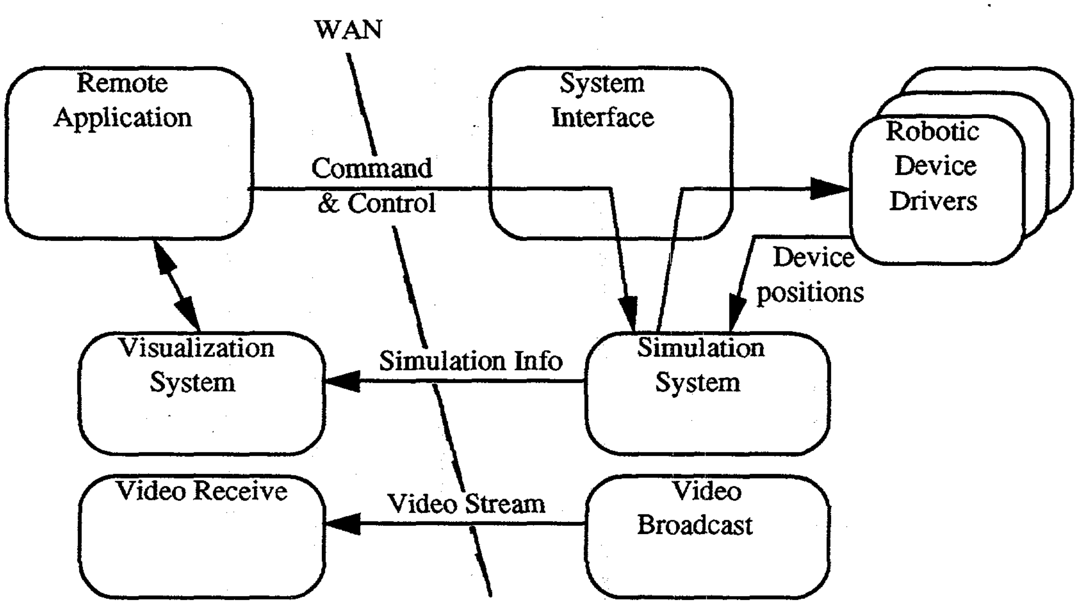

Figure A-4. Local Operator as Filter Architecture.

Brief Description

The remote operator uses a remote supervisor application to define and plan tasks for the robot system. The remote operator may use simulation software to automate plan generation. Operation plans (i.e., those plans that perform a task) are transmitted to a local system interface for safety analysis and viewed graphically through a remote simulation visualization system. Both the remote and the local operator accept safe and effective plans. Accepted plans are executed in hardware and monitored graphically and visually through the remote simulation viewer and remote video systems.

The local operator manages the system interface to simulate requested tasks, add unmodeled constraints, and approve or reject safe or unsafe tasks plans. The local operator is responsible for maintaining the accuracy of the world model and the simulation environment, for properly using and maintaining the system interface program, and for monitoring simulations to ensure that unmodeled constraints are adequately managed. The simulation system automatically rejects requests that would create known safety problems such as causing a collision or over-running a joint limit. The local operator's user interface allows the operator to accept or reject all other motions.

Features

All motion requests in the Local Operator as Filter architecture are approved at a minimum by the local operator (either automatically or manually) and optionally by the remote operator. This approval process includes safety simulations that may duplicate simulation performed in the planning process. The local operator has ultimate authority for accepting or rejecting all motion requests and is appropriately empowered and informed about the proceedings.

It is the hardware provider's responsibility to determine how accurate or expensive the simulation system should be. The sophistication of the simulation system sophistication, however, does not impact the remote user's system requirements. For example, expensive systems can be teamed with expensive hardware and inexpensive systems with inexpensive hardware without affecting the remote interface.

Advantages

Placing the validated simulation system at the local operator's control has many advantages over placing it at the remote operator's console, as follows:

1. The remote user can use a variety of software to command the hardware. This provides a higher degree of flexibility and allows the remote operators to develop their own software without degrading safety.

2. The local operator can maintain a trusted world model without concern that the remote operator may inadvertently modify the model in a way that loses its validity. Trusted models are more efficient to use because they do not need to be continually questioned.

3. Model asymmetries can be used to allow efficient safety simulations to be carried out at a local system while more visually oriented systems or alternatively lower-cost systems can be used on the remote side.

4. Model asymmetries can be used to allow the remote and local viewers to monitor the system differently. For example, the local system might emphasize efficient safety simulations while more visually oriented systems or alternatively lower-cost systems can be used on the remote side.

Disadvantages

The remote operator does not have direct control of the hardware and must always work through the intermediary local application system. This indirection can cause increased delay between defining a task and executing it in hardware.

The remote system must replicate the geometric, kinematics, and behavior models to perform task planning with the remote system. Otherwise, all task planning must be done by the local system. (Remote task planning is useful when the remote operator wishes to manage or hide proprietary process information from the hardware provider.) Duplication of behavior models results in increased system complication. The duplication may also result in duplicated simulation steps (once in planning, again in safety checking) and slow system use.

Distributed Collaborative Architecture

The Distributed Collaborative architecture, shown in Figure A-5, distributes responsibilities across users according to their expertise and location. This architecture is a logical extension of the two prior architectures. Planning and simulation steps are still performed but activities and responsibilities are better matched to operator location and capability.

All users' systems share information as it is generated. Command and control structures are maintained to ensure machine safety and single point of control. Multiple users share system control at levels of abstraction that are appropriate to their needs. Thus, for example, robot motion may be communicated to an assembly Computer-Aided Engineering (CAE) system as component assembly motion.

Possibilities for implementing this concept are unlimited. Probable development would follow deployment of systems of the two prior architectures in an evolutionary fashion.

Brief Description

Each user or collaborator has a software collaboration system that represents world-model data in ways that are appropriate to their work. These collaboration systems typically include planning, simulation, and analysis components. They communicate with other collaboration systems at object levels to transmit commands and information.

When support is needed, collaborators link their systems to allow the requester's system to send requests in the form of commands to the provider's system. From the commands, the provider develops and simulates plans that fulfill the request. Both live and cumulative simulation data is shared with the requester who determines whether the plan meets known requirements. Requests which require lengthy analysis may be communicated in batch form.

Collaborative activities that require several levels of refinement may or may not link commands in real-time through each level of task collaborators. For example, a designer may wish to transmit an analysis request to a finite-element simulation provider and then disconnect and link to a manufacturing planner who in turn links to a manufacturing cell provider while retaining the link to the designer.

Simple Robotics Example

The remote user uses a simple graphical display to define tasks for the robot. The graphical display contains a video snapshot of the workcell and various task buttons. The user indicates the task by first selecting a menu option and then specifying the location for the task on the video image. The task request is then transmitted to a robot operator's system. The local operator's task planner uses its correct world model and simulation environment to generate and simulate a set of operations for achieving the task.

The remote operator views the task being simulated through a graphics sharing system or a video broadcast system, or from a generated movie. Alternatively, the simulation results could be used to drive the visual interface to show the expected effect of the operation (e.g., draw a patch showing what parts of the image would change).

At the concurrence of the local and remote operator, the mechatronic system would be run, process parameters would be recorded, and new images of the transformed worksite would be transmitted. The local operator's world model would be updated as necessary and subsequent operations would be performed in a similar way.

Factory of the Future (FoF) Example

This architecture closely follows FoF concepts while avoiding the ubermensch phenomenon.[9] Responsibility and expertise for executing components of tasks is highly distributed but information generated is highly integrated. This example shows how the FoF system would function.

An engineer generates a design for a component with a CAE system and transmits the design and a request for analysis to a manufacturing engineering (MfgE) collaborator. The analysis uses science-based design technologies and human input to both generate a manufacturing plan and update the CAE system for probable component performance. Manufacturing analysis data is also used to locate appropriate existing mechatronic systems and to determine probable costing data. The designer redesigns if the probable cost, delivery and performance are not acceptable or continues on if they are.

Next, a mechatronic preprocessing system is selected and manufacturing plans are transmitted to the factory planner. Operations are converted into machine primitives and simulated and checked for safety and efficacy. The MfgE is consulted to develop alternative plans if they are needed. Acceptable plans are transmitted to the mechatronic system for preparation.

Materials and fixtures are then loaded into the mechatronic system, and the mechatronic system world model is updated with these changes. Operation plans are fine-tuned using the real locations of the materials and fixtures and the real starting and requested final condition (position) of the mechatronic system. Operations are simulated for final safety and efficacy and changes (mainly for fixture and material locations but possibly all the way to the design) are made as required.

Finally, the workcell is run and real-time data is collected both during processing (for process parameters) and after processing (for inspection). This real-time data is transmitted and refined by each successive planning module by comparing actual with simulated data. Fully processed information is submitted into the CAE system for science-based analysis of part reliability and quality.

---

[9] The overman (Ubermensch) is Nietzsche's concept of a human being who has created for himself that position in the cosmos which the Bible considered his divine birthright. "I teach you of the overman. Man is something that shall be overcome." Nietzsche, **Thus Spoke Zarathustra**, 1891. Some efforts to create systems that a single, all powerful user controls from design to production seem to have their genius with Nietzsche. "I love him who works and invents to build a house for the overman and to prepare earth, animal, and plant for him: for thus he wants to go under."

Features

Collaborators of systems communicate at both the command and information level. The collaborators also are likely to communicate by voice and other means to convey concepts not contained in the computer models and commands. Information sharing is a human interactive task while the computer systems automate information exchange.

Advantages

The Distributed Collaborative architecture is a way-cool concept. Collaborators get to use the resources of others without having to learn their systems. And it's realistic. Each collaborator is put in control of what he/she ought to control.

Disadvantages

The Distributed Collaborative architecture is tough to build and it's complicated. We've never done anything like it. The remote operator may not find it easy to determine if the simulated task is effective by monitoring only the remote system. The system might be frustrating to use.

Some potential customers won't get it. No single user will see the whole system. Other users might seem to customers like wizards behind curtains ("but what part is automatic").

Information Needs

All the systems described in this report share similar information needs. The information that the VCE operators and users must share includes geometric, property, state, task, sensed, and person-to-person data.

- Geometric information includes positions of objects in the work environment, kinematic models, and planned motions (trajectories).

- Property modeling information consists of names, physical properties including mass and composition, and relational properties including part hierarchies.

- State information pertains to the system's current mode of operation (including tools held) and control and tool state (on/off, lock/unlock).

- Task information includes information about the task being set up or performed, but it also includes task constraint and validity information that might be embedded in the model. For example, a pipe might "know" that it can be cut with a pipe cutter.

- Sensed information covers the sensed geometric positions of the objects in the environment as well as the sensory data such as pictures, measured radiation levels and measured surface finishes, and forces of interaction that cannot be easily represented as model geometry.

- Person-to-person data includes the voice and picture data that the various users might share. While easily overlooked, the interpersonal communications are critical to effective operation and designing these communications into the system may provide the biggest leveraging issue of VCEs.

Person-to-person information sharing encompasses a variety of needs. People must communicate about what they want to have done and about what they are doing. Verbal and visual exchanges are continually required. Collaborators are also likely to need to communicate or share computer interfaces to show why and how (in their own way) certain constraints must be maintained.

Geometric and property information must be transmitted in the form of files with models and as model parameters. Model asymmetry must be supported to allow different users with different needs to link parts of their models with each other. Registration of asymmetric models must be supported.

Task or command and state information must be transmitted in a reliable way. Communications technologies for the transmission of robot control commands are appropriate for this type of information exchange.

Sensed information can be communicated through reporting mechanisms of command and control systems, through batch reporting and file sharing systems, or through real-time communications that preserve temporal integrity.

These information needs are supported by the variety of technologies described in this report. Establishing interfaces that support the technologies will allow development through the most sophisticated VCE technologies proposed here.

Appendix B: Potential-User Evaluations Report

During the first two years of this LDRD project, Sandia solicited feedback from potential users of the Virtual Collaborative Environment (VCE) technology. User assessments of the new technology were solicited through visits to customer sites, demonstrations at trade shows, presentations at conferences, and technical papers. This appendix discusses the issues communicated to potential users during the evaluation process and summarizes the feedback obtained from them. Please note that this appendix was written in mid-1996 and reflects the status of the LDRD project at that time.

Statement of the Problem

Increasing pressures to do more with less have stimulated innovative ways to share resources and to perform team-based technology development. While software-resource sharing has grown with the proliferation of enabling electronic network technologies, network-based sharing of electromechanical equipment is uncommon. A VCE is a system for sharing mechatronic technologies (e.g., robots or machine systems and associated software) among a group of participants. The basic VCE approach is analogous to how multiuser computers are shared among a broad group of users. In a multiuser computer environment, the computer is brought virtually to the user's desk by redirecting the computer's data streams (e.g., a terminal is a virtual connection to the operator's console). In a VCE system, mechatronic resources are brought to the desks and laboratories of users through virtual representations of the machines. The users interact with the virtual machines while the resulting computer programs run both the virtual and real machines.

Approach

The purpose of this report is to describe an interactive virtual collaborative testbed that will help to stimulate the widespread sharing of mechatronic resources within the United States. Sandia will serve as the core site to initiate this new approach to technology development. Sandia will provide a network-based server node for software libraries, access to state-of-the-art robotic and other mechatronic hardware and software for testing and technology development, and coordination of resource-sharing activities among universities, national laboratories, and industry. The proposed testbed will be used for sharing mechatronic technology resources. Collaborators can reside at distant locations while linked electronically with the shared mechatronic testbed. Machine systems that are located and maintained at dispersed sites can also be accessed and operated as part of an integrated distributed system.

The proposed core site at Sandia is being networked to other institutions that are under contract to DOE to perform research supporting the use of robots in hazardous environments. These institutions include the National Institute of Standards and Technology (NIST), Oak Ridge National Laboratories (ORNL), Pacific Northwest

Laboratories (PNL), Los Alamos National Laboratories (LANL), Lawrence Livermore National Laboratories (LLNL), and universities such as Carnegie Mellon University (CMU). Access to the mechatronic resources will be made through the Sandia gateway as these resources come online.

Potential users of the VCE testbed may be familiar with the actual testbed sites or may have visited them only in a virtual sense. A virtual representation of a real system consists of real-time graphical models and live video images of an actual site. The VCE testbed allows the users to interact with virtual systems at a very abstract level, tests the users' plans against safety and other constraints, and executes experimental test plans on real systems by delivering experimental data in near real-time to remote experimenters. Initially, the robotics facility at Sandia will be used for the VCE testbed. In time, mechatronic resources such as those at NIST and other institutions will be integrated with this testbed environment to provide a national testbed. Because the testbed enables the use of mechatronic systems by diverse users, an information architecture that transmits and uses information to automate programming of machines is an integral part of the VCE concept. As new mechatronic technologies evolve through innovation, testing, and integration, the VCE testbed will become a repository for mechatronic technologies. Technology insertion is thus accelerated as a wide range of implementers access the repository.

The Impact of the VCE Testbed

Economic competitiveness in the future will be very dependent on aggressive technology development coupled with rapid insertion into marketable products. Teaming to reduce the cost and time for technology development is a key strategy. The VCE testbed accelerates technology development by taking advantage of the rapidly growing national information infrastructure and coupling information directly to the operation of integrated systems of machines in a state-of-the-art testbed. A basic motivation for using VCE technology is that information can be transmitted electronically as computer models whereas equipment cannot. In addition, highly modular software technologies with well-defined interfaces allow rapid synthesis of large integrated software systems without requiring that large amounts of code be written. Software modules written by different developers can be interfaced with each other through the electronic network that forms the backbone of the testbed.

Before deployment, new technologies must be evaluated using actual machines and systems. The VCE testbed provides access to widely distributed resources in the form of computer models and data and then couples these virtual technologies to operating equipment at Sandia (and other sites as available) for testing and production. Thus, technology development will be increasingly accomplished within virtual environments followed by testing in remote state-of-the-art experimental facilities to verify operational characteristics. Access to state-of-the-art mechatronic systems at Sandia for technology development significantly reduces the capital costs for innovators while providing realistic experimental environments.

A technology developer or product designer might access national resources over high-speed electronic networks and test new concepts and/or designs against virtual models of an enterprise or its components. This same network would then allow the technologist to test new concepts on operational state-of-the-art equipment. The results of those experiments could also be used to improve the fidelity of the distributed modeling resources, thus supporting evolution to higher levels of capability and forming the framework of a nationally accessible repository for technology.

The VCE approach described above thus makes limited state-of-the-art software and hardware mechatronic resources available to innovators, both large and small. Widespread use of such resources through computer models is facilitated by sharing, rather than by directly interacting with expensive machines and test equipment. In many respects, we are starting on a journey that will lead to very-large-scale sharing of resources. This sharing will derive from the movement of information across the proposed National Information Highway because machines cannot be moved easily from site to site. However, the testing of new technologies using actual machines is critical to successful technology development.

Summary

Information-based approaches to system integration allow the sharing of both software and hardware resources for collaborative technology development. This distributed technology development environment reflects the basic structure of the distributed manufacturing environments envisioned for agile manufacturing. This environment accelerates use of the new technology because much of the controlling software can be directly compiled for use within distributed manufacturing scenarios. In addition, a technology developer can work at the system level without incurring the costs associated with constructing expensive hardware test environments. Thus, innovators from small and large organizations can interact and contribute their ideas and innovations with minimal embedded costs. Accelerated technology transfer and insertion occurs when industry, university, and government laboratory use a common testbed environment and identify those technologies that enhance their own work and that have the most commercial promise.

Proposal Response

During fiscal year (FY) 1995, development of the VCE testbed was funded by the Advanced Research Projects Agency (ARPA), DOE's Robotic Technology Development Program (RTDP), Sandia's Computing Center, and the LDRD program, which is Sandia's internal research and development funding mechanism. During FY 1996, the overall level of funding for the VCE testbed has been somewhat reduced, however VCE development has become a major focus in RTDP.

Sandia's association with NIST (Joe Falco) and the University of New Mexico (Ron Lumia) during the various funded projects has resulted in a commitment by each organization to configure a VCE-accessible site. NIST is implementing a hexapod workcell for access via an IGRIP-SGI VCE system. UNM has obtained a PUMA 562 robot and is installing a Cimetrix controller for access via the low-cost user interface being developed by Sandia.

Conference Demonstrations Response

The VCE technology has been successfully demonstrated at several national conferences. Of particular note are the demonstrations conducted during the Robotics and Vision Automation Show held in Detroit, Michigan in May, 1995. More than 210 individuals representing a variety of industries, manufacturers, universities, military, research labs, and other government agencies sought additional information about the VCE technology and other Sandia-developed technologies showcased at the conference. Sandia distributed a letter and conference brochures to each visitor who expressed interest. The letter included the address of a web server where additional information about the testbed was available.

In general, participants in the conference demonstrations expressed interest in the VCE technology, but they were concerned about the costs associated with the access infrastructure. The costs of IGRIP software, SGI workstations, TI wide-area-network connections, and Sun videoconferencing products are an order of magnitude too high for most potential users. Participants also expressed interest in using off-the-shelf products that were commercially supported and based on industry standards. This feedback resulted in the decision to develop a low-cost VCE access system based on the IBM PC computing platform during FY 1996. The IBM PC is an industry standard and it is readily available at a reasonable cost to potential users.

Visits to Potential User Sites

During March and April of 1995, VCE developers visited several potential users at their respective facilities. The purpose of these visits was to elicit comments and suggestions about the VCE testbed. Presentations describing the VCE testbed were made at each site and a lengthy discussion with the potential users followed. The list below represents a sampling of the feedback obtained from a variety of users and industry experts at the selected sites.

1. Some potential users view virtual manufacturing and VCE as primarily comprised of simulation and modeling. Manufacturers typically work with compatible system components. That is, they purchase production equipment that is operable with other workcell components.

2. In manufacturing, many question the usefulness of Off-Line Programming (OLP), except in a few applications such as painting, which represents less than 10% of applications. Many manufacturers perceive that (1) the technician or person on the line who is most familiar with the part process should create the robot path because expert systems cannot do the job, (2) the investment and problems associated with modeling the workcell and robot do not justify the benefits, (3) programming the geometry (or teaching points manually) is simple, and (4) the robot is too inaccurate in most cases to perform the OLP-generated path adequately.

3. For the limited situations where simulation might be useful, one company contracts out the work to others.

4. Overall factory simulation is perceived as more useful by one company than robot simulation, given the inaccuracy of robot simulation.

5. Many potential users have negative opinions of OLP, based on past failures with this robot programming method.

6. Problems attributed to the reliability of robots are associated with end-of-arm-tooling, communications, and support-related issues. These problems are not associated with the robot itself. In fact, robots are very reliable (less than 5% of the robot system downtime attributable to the robot). A typical robot system requires human intervention at least every two hours because of a failure in some part of the system. Future competition between robot manufacturers may be based on costs associated with robot procurement, robot system spare-parts inventory, and maintenance.

7. One company expert suggested that a possible application of VCE technology was to use a real sensor for driving the simulated robot. In response, VCE developers proposed a scenario to configure a machine-vision controlled robot system in the VCE environment, and then work out the control code using a machine-vision sensor for controlling a virtual robot before actual hardware implementation.

8. One industry expert estimated that 10% of a robot program was geometry-related (taught points) and that the remaining 90% was control-related (e.g., velocity, move type, error handling, and open/close end-effector (EE). This expert thought that the aerospace industry might be interested in the Distributed Collaborative Workbench (DCW) technology because the per-part costs were much higher and the number of parts were lower. In addition, the CAD information is generally richer in the aerospace industry.

9. PC-based (preferably Windows-based) programming systems for robots have the best chance for acceptance and use by manufacturers, who are looking for

the easiest way to program the robots. Many low-level programming tools are currently being used. Although such tools do not have fancy graphics, they are easy to use. Point-and-click interfaces and robot-independent programming (statements such as MoveTo are automatically translated into the host robot's language) are highly useful.

10. Automatic path planners are basically a solution looking for a problem. In manufacturing, paths are obvious and easily created manually.

11. Robot calibration is sometimes performed by applying a calibrated robot model in OLP software to resolve the inaccuracy. Points do need additional modification from time to time. One method that is being pursued for improving the calibration and positioning accuracy of robots is to use a 100 Hz laser-tracking interferometer for controlling the end points.

12. One industry expert estimated that 60–70% of the robot applications in his company are programmed offline, 25% with teach pendants, and less than 5% are sensor-driven. This expert concurred with experts from other companies that the robots are very reliable but that the peripherals and communications generally cause robot workcell downtime.

13. Aerospace companies tend to invest heavily in fixtures—some fixtures take over one million man-hours to build. Research is being conducted in adaptive fixturing at one of the companies visited.

14. In the automotive industry, robots are finally living up to their expectations. The controllers have improved dramatically, and robots provide highly reliable services at attractive prices without competing with people. One of the ways that the automobile industry is getting lean is by using robots well.

15. An expert in the automotive industry estimated that robot applications at his facility consisted of 51% spot-welding, 7% painting, 6% material handling, 4% assembly, and the rest in electronics fabrication (electronics fabrication generally employs vision; otherwise no vision is used). Simulation is used by one-third of the people in the robot department of this facility. However, simulation is challenging for the following reasons: a) there are calibration problems between the model and the simulation, b) calibration is costly and clumsy to perform, and c) models are hard to acquire and either incomplete or inaccurate.

16. An automotive industry expert has found that simulation works well for laying out and designing a workcell. Simulation is perceived as a means of graphically demonstrating robot product operation but generally doesn't add significant value. Using OLP for painting and sealing applications adds significant value because of the complexity of programming and the number of

target points required. Generally, if an application has only 10–15 points, then teach pendant programming is the best approach. If an application has 50–100 points, then OLP methods and perhaps even simulation are appropriate.

17. One industry expert suggested two strategies for determining the relevance of simulation. The first strategy is to find out why industries are not using simulation (e.g., inherent weaknesses in simulation products like slowness, insufficient accuracy, high costs, and wrong emphasis). The second strategy is to approach manufacturers and ask them to identify their real needs for simulation.

18. Some companies have launched very aggressive initiatives to merge their business, engineering, and design databases. Such initiatives enable the companies to relate information and access data in useful ways.

19. One automotive company uses extensive simulation, has its own automatic path planner with collision avoidance, and performs a two-part calibration on robots to enable OLP. The first calibration is a robot footprint calibration to determine the actual Denavit Hartenberg (DH) parameter set, and the second calibration is a rigid-body correction to align the robot with the world coordinate frame. The company claims to have achieved near-repeatability levels of accuracy after performing this two-part calibration.

20. Recently, a company in the automotive industry installed 250 robots in one of their plants; 170 of the robots were programmed offline. The robots are used in spot-welding (80%), material handling (10%), and sealing/painting (10%). Generally as many as two vision systems are employed in a windshield-placement operation.

21. Sensor-based control is used infrequently by one automotive industry company. However, the company does use optical sensor-calibration systems to recalibrate the location of the terminal control point in a spot-welding gun.

22. Some companies in the automotive industry expressed interest in modeling machine code operation and the logic of processors. Debugging of logical processes can take "weeks and weeks," with validation identified as a big problem. The companies need simulations for each robot language they use, but they cannot perform true simulation of machine tools. These companies are also interested in having the ability to model hoses, cables, bending forces, and other kinematic restrictions caused by bending hoses and cables. They are working on optimizing their path planner to avoid collisions while minimizing cycle times.

23. One automotive manufacturer is modeling human operations and calculating energy expenditures and lift indices. In one such simulation, a tire-loading

operation demonstrated that a person could not eat enough calories to sustain the operation for 8 hours. This analysis partially explained why the company has experienced extremely high turnover rates for the tire-loading position.

Response to Technical Papers and Conference Presentations

Response to technical papers and conference presentations has been limited. Although technical reviewers have noted that the technology is very exciting and dynamic and that the papers are well-written, there have been no requests from readers for more information. A couple of potential users from universities have expressed interest in the technology during presentations of technical papers, but none of these contacts have resulted in serious discussions about further collaborative work.

DISTRIBUTION

5	1080	B. Davies, 1325
5	1008	M. J. McDonald, 9621
1	9018	Central Technical Files, 8940-2
5	0899	Technical Library, 4916
2	0619	Review and Approval Desk, 12690 for DOE/OSTI
2	0188	LDRD Office, 4523