INTELLIGENT SYSTEM CONTROLLER FOR REMOTE SYSTEMS

R. W. Harrigan
Sandia National Laboratories

ABSTRACT

The U.S. Department of Energy's Office of Technology Development (OTD) has sponsored the development of the Generic Intelligent System Controller (GISC) for application to the clean up of hazardous waste sites. Of primary interest to the OTD is the development of technologies which result in faster, safer, and cheaper cleanup of hazardous waste sites than possible using conventional approaches. An objective of the GISC development project is to achieve these goals by developing a modular robotics control approach which reduces the time and cost of development by allowing reuse of control system software and uses computer models to improve the safety of remote site cleanup while reducing the time and life cycle costs.

INTRODUCTION

The concept of a Generic Intelligent System Controller (GISC) resulted from a series of joint DOE/NIST workshops initiated in January 1990 [1,2] These workshops were held to determine what generalized approaches should be employed in the development of control system structures to support development of robotic systems for application to DOE waste cleanup problems. The goals of the workshops were to identify a <u>system</u> control environment which would

1. stimulate cooperative team based robotics system development,
2. facilitate the rapid fielding of integrated robot system demonstrations, and
3. stimulate common approaches to system control within the Robotics Technology Development Program (RTDP).

The consensus of the first Workshop was that modular open architectures to stimulate subsystem integration should be stressed. The second workshop identified the importance of multiprocessing computing environments for complex system control. The Unix operating system was identified as the operating system preferred for software development and VxWorks was identified as the preferred operating system for the real-time subsystem control environment. C and C++ were identified as the preferred programming languages. In addition, the use of graphics operator interfaces to simplify system programming was determined to be highly beneficial. Finally, the need to employ architectures and environments which could easily evolve to take advantage of improvements in computing and robotics technologies was stressed to prevent rapid obsolescence. It was felt to be very important to provide architectures which would both facilitate software development (thus the Unix based environments) and support the real-time multiprocessing computing environments (VME bus, MC68000 series single board CPUs) needed to support the development of complex robotic systems.

MASTER

A GENERIC INTELLIGENT SYSTEM CONTROLLER APPROACH

The emphasis in the development of the GISC approach was functional not architectural. As a result, GISC is a strategy for integrating diverse subsystems. The emphasis, following the results of the NIST/DOE Workshops was on development of modularity and extensibility. There was an emphasis on using commercial technologies as much as possible and to extend these technologies to facilitate robotic system development. At all times during the development of the GISC approach, a goal was to produce a robotic system control environment which resulted in facilitating use of robotic technology by the site remediation technologists. As shown in Figure 1, GISC coordinates and integrates the operation of diverse subsystems to accomplish complex tasks. GISC is a modular approach for combining robots with sensors and computer models to provide robotic systems that are automatically programmed and much safer than conventional remote systems.
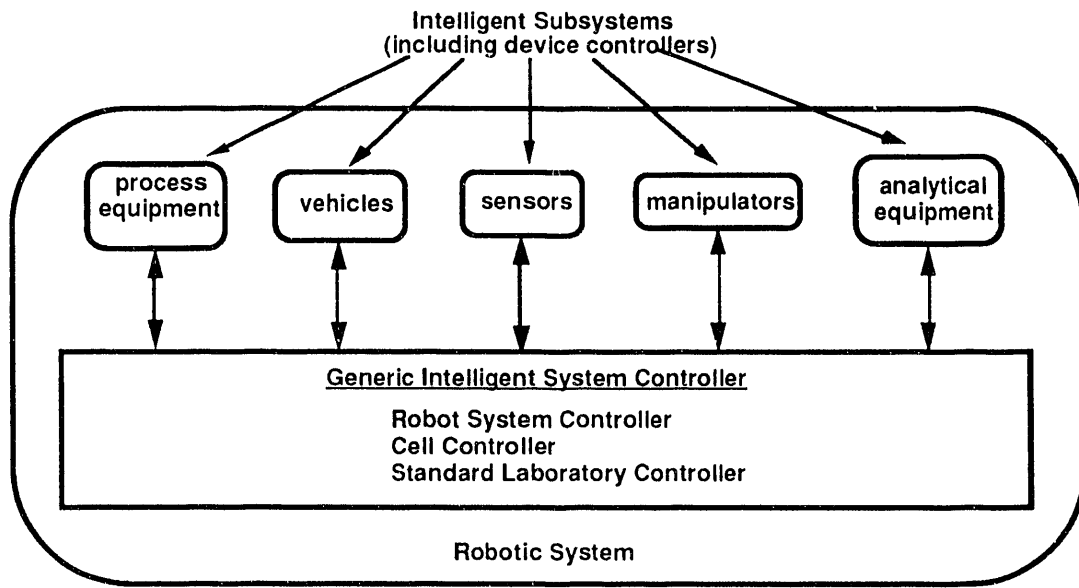

Figure 1
GISC Integrates Diverse Subsystems

GISC is termed an intelligent control system approach because computer models automate much of the system programming and condition the operator interfaces to facilitate correct decision making by a human supervisor. In addition, computer models monitor all operations (both computer planned and operator initiated) for safety. Algorithms evaluate the known environment represented by a World Model, plan collision free robot motions, and then automatically generate the robot command sequences necessary to execute the desired robot motions. Use of the computer models for error detection and recovery prevents unsafe actions which might, for example, result in collisions. The incorporation of computer models into GISC's approach to system control also enables the use of visualization technologies to provide intuitive graphic operator interfaces that allow even unskilled operators to safely program complex tasks.

Figure 2 recasts the diagram of Figure 1 in terms of the major functional blocks of GISC. As indicated, the GISC environment communicates with multiple intelligent subsystems through their respective controllers. Extensive use of computer models (labeled World Model in Figure 2) and sensors allows both operator directed task executions as well as automated operation. Direct operator control of

individual subsystems is allowed only on rare occasions during emergency situations or during system development for debug purposes.
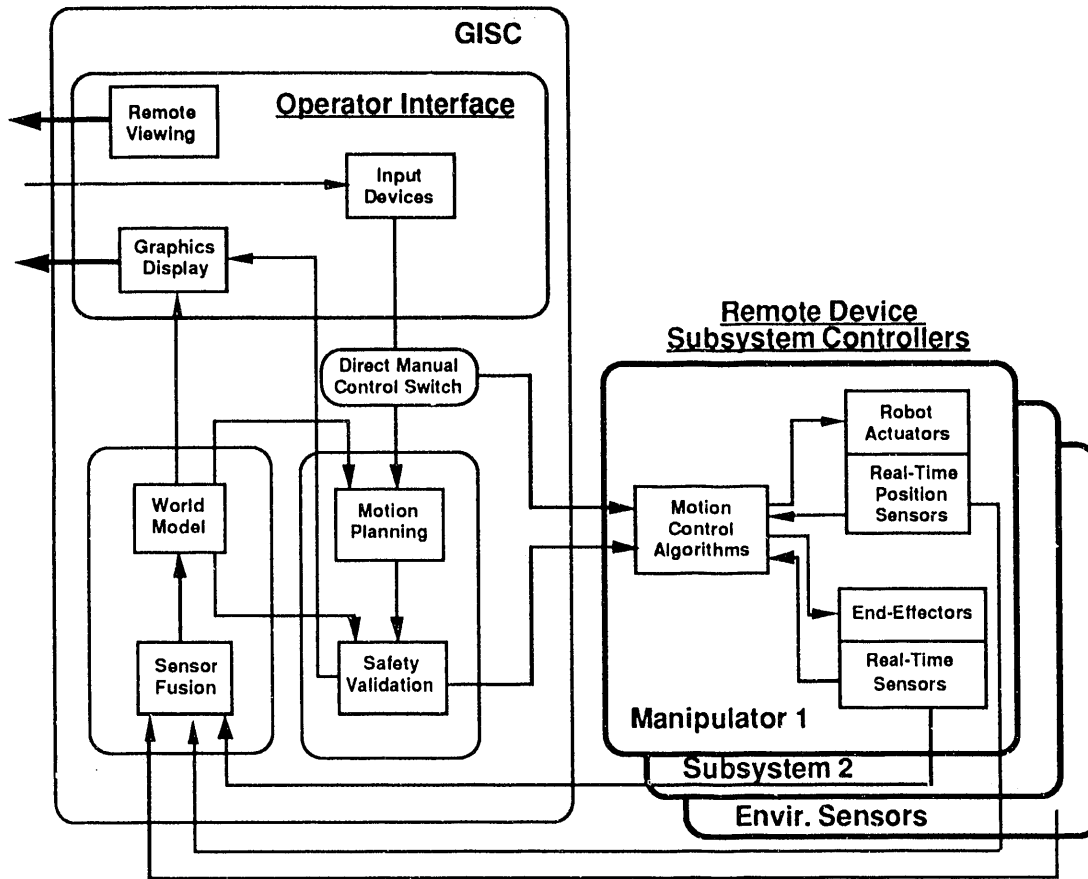

Figure 2

Generic Intelligent System Controller

The World Model is at the heart of GISC. The World Model contains the information about the robot system and its environment needed to monitor system operation and to plan the execution of tasks. The World Model is developed by combining engineering knowledge of the overall system including information from engineering drawings about known structures in the environment, robot end effectors and tools, knowledge of proper operating procedures, and knowledge of the robot system kinematics. While CAD information of the robot's environment together with the kinematic and dynamic models of the robot and other devices typically make up the major portion of the World Model, sensors also provide information to the World Model. Sensors both determine new information which is integrated into the World Model and provide updates to the World Model as operations are executed. The World Model is thus dynamically updated at all times based upon the most recent sensor information.

Visualization software technologies provide the operator of the robot system with animated graphic displays of the computer's knowledge base allowing the operator to verify that the World Model is correct and to preview intended robot movements prior to actual execution. In a way, visualization allows the operator to

view what the computer is *thinking*. The close linkage of the World Model with the graphics interface allows all programming to be done in the graphics environment. Operator directed movement of devices within the graphics interface are automatically compiled into robot commands using information contained in the World Model. After graphically programming a desired sequence of robot movements, the intended programmed movements can be reviewed and modified by the operator or communicated to the robotic device for execution. At no time will the computing system allow execution of a programmed operation if the World Model determines that unsafe operation may result.

As shown in Figure 2, all operator commands to the system are checked against the knowledge contained in the World Model to ensure that the command can be executed safely and within any Standard Operating Procedures. If the operator's command can be executed safely, the command is communicated to the subsystem Motion Control module. In the case of Figure 2, this is the module which contains the real-time system servo control algorithms for movement of the remote robot subsystem. High speed sensing subsystems provide sensory inputs to the robot subsystem motion control algorithms if necessary to perform, for example, force controlled interactions with the environment. Real-time sensing also allows in-operation modification of the robot system motions to, for example, prevent collisions with previously unknown obstacles or to allow the robot end effector to track the surface of the waste. It is important to recognize that such high speed sensor based servo controlled operations are executed within the subsystem controller and not by the higher level GISC environment. This prevents delays due to bottlenecks in communication and computation. Distributed computing environments are an important feature of the GISC approach to robotic system control. This provides natural modularity and extensibility and enhances the responsiveness of large robotic systems.

Sensors used for real-time control of discrete subsystems communicate with the higher level GISC models to provide dynamic updating of the World Model. The updated models are used both to detect possible safety and operational problems as well as provide the accurate World Models needed for automated planning and programming of new operations. The importance of communicating dynamic sensory information to the GISC World Model must be stressed. Since the World Model is used to validate all commands to the active subsystems, it is important that the World Model contain measured information wherever possible. Use of measured information such as manipulator joint angles will not only provide updated World Models but allow validation that the computer models used by the World Model to automate operations accurately reflect the motions generated by the subsystem controllers. This is especially true during sensor based operations when the motion of the manipulator is governed by real-time sensor inputs not precomputed trajectories.

If an operator's command to the robot system is deemed unsafe when examined with respect to the knowledge contained in the World Model, the operator's command is either modified slightly to, for example, allow the robot to avoid a known obstacle, or interrupted completely. If the operator's command is interrupted, the operator is alerted and an indication of why the command is thought to be unsafe is sent to the operator (usually via the graphics display). One mode of operation is to display the results of all commanded robot operations graphically to the operator prior to the operator issuing the final permission for the robot system to execute the operation. This is an important operational mode since it is not always clear to an operator what the result of a commanded motion might be. Joint limitations in some commercial manipulators, for example, can result in unanticipated motions to achieve

what to the operator is a very simple move command. Such unanticipated motions could result in collision if not detected prior to execution.

Notice that to this point, the description of the control system architecture is very generic. There has been no discussion of the specific robot used in the waste recovery operations nor of the characteristics of the waste environment. The general control system architecture allows variations simply by changing the knowledge contained within the World Model or the Motion Control modules. As our understanding of the problem changes, the detailed nature of the individual control system modules may change but, the basic GISC structure remains the same.

APPLICATION OF GISC

A first implementation of GISC has been completed and is currently in use at Hanford, Washington as part of the Underground Storage Tank Robotics Technology Development Program. The tasks currently being addressed by the OTD's Robotics Technology Development Program (RTDP) are mapping of internal structures and waste surfaces in underground storage tanks (USTs) as well as removal of the waste and structures such as piping from the tank both for laboratory analysis and tank cleanup. Both fully automatic and manual robot control technologies are being developed and demonstrated [3]. Computer assistance is employed whenever manual control of the robot is required.

The UST Robot Technology Testbed employs a large commercial Spar Aerospace robot as a high strength positioning system for a dextrous Schilling Titan manipulator. A Remote Tank Inspection (RTI) Robot developed by RedZone Robotics was used to deploy sensors for tank wall inspection operations. Pacific Northwest Laboratory developed a Tracking Camera which, using information from the robot's joint encoders could automatically track the motion of any part (typically the end effector) of the moving manipulators. Both ultrasonic and optical Proximity Sensors were used for localized waste surface detailed mapping and position control of the Titan manipulator. Structured Lighting was used to map the waste surface for addition to the World Model prior to the start of robotic operations. Engineering drawings of the testbed served as the Apriori Engineering Data for the initial World Model. The multilaboratory project team members included Westinghouse Hanford Company, Oak Ridge National Laboratory, Pacific Northwest Laboratory, Idaho National Engineering Laboratory, Sandia National Laboratories, as well as representatives from Spar Aerospace and RedZone Robotics [3,4,5].

The computing structure of this first implementation of GISC is shown in Figure 3 together with the ethernet communication interfaces between GISC and the intelligent subsystems. In all cases the primary interfaces are Motorola 680X0 series CPUs in a VME backplane. These CPUs interpret the generalized GISC commands and convert them to the commands understood by the individual subsystem controllers. In addition, these VME based interface units also interfaced the ethernet communication system with the subsystem communications requirements of the specific robotic devices (e.g., RS232 serial communication in the case of the Spar). The VxWorks operating system was used on the VME systems. The equipment outside the box marked GISC represent independent subsystems with diverse computing and control environments. In one case, the Tracking Camera, an IBM PC/AT was used as the subsystem controller and serial communication was used to interface this subsystem into the overall system. The modularity of the UST Robotics Testbed is apparent as is the ability of GISC to integrate diverse subsystems into a coordinated system addressing key waste cleanup issues.

Figure 3
UST Robotics Demonstration -- GISC

Figure 3. UST Robotics Demonstration -- GISC
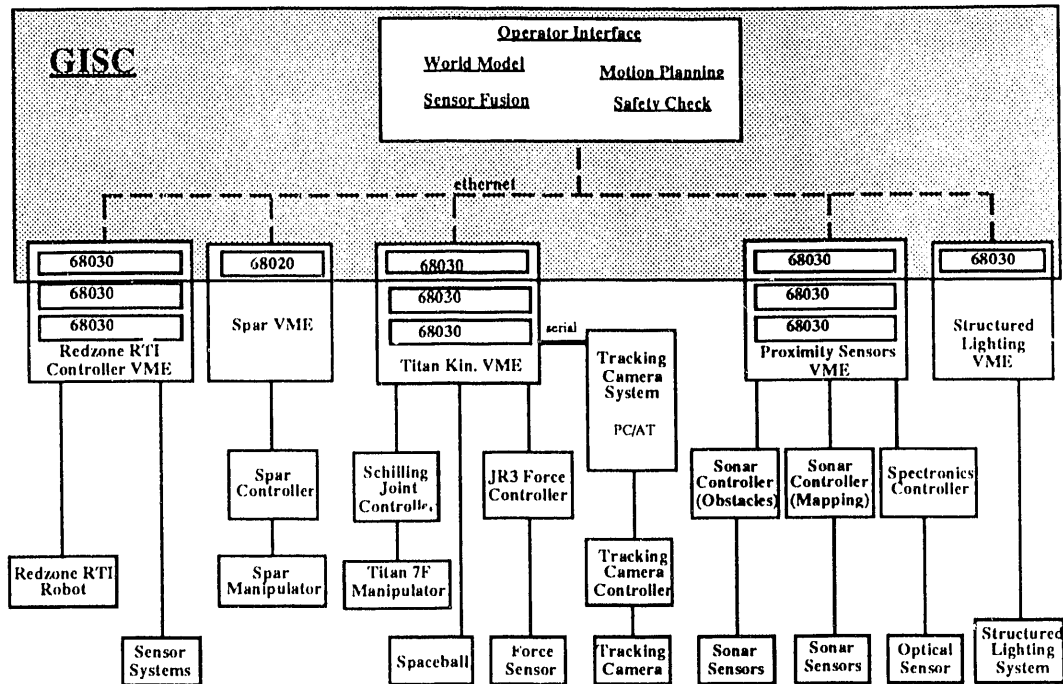
The Proximity Sensors subsystem performed two functions. In the first, sonar proximity sensors were used to provide high precision maps of the waste surface and vertical piping (risers) within the testbed environment. The Proximity Sensors module performed the statistical correlation of the multiple sonar sensors with location of the Titan manipulator (upon which the sensors were deployed) to compute map information locally to the Proximity Sensors subsystem. This computed map information was then communicated to the Sensor Fusion module of GISC and entered into the World Model. During proximity servo control of the Titan manipulator, range information was passed directly to the Titan subsystem controller without direct GISC involvement (except for the communication software) to allow high speed servo control. Proximity information for servo control could be obtained either from the sonar sensors or the Spectronics optical sensor, both of which were being tested.

The Structured Lighting used a laser based sensing system to develop maps of the waste surface to enable the robot subsystems to safely approach the waste surface for more detailed operations. The Structured Lighting subsystem analyzed all optical data locally and passed only processed map information to GISC to be combined with the initial Engineering Data to provide more complete World Models for automating robot system programming. Thus, the modular structure desired as part of GISC was implemented in a straight forward manner. The modularity enabled multilab teams to work easily together. System integration for this complex systems required less than 10 weeks.

Results

GISC and the associated subsystem controllers have been implemented and first system operation as part of the Hanford Underground Storage Tank Robotics Testbed was demonstrated in August 1991. The World Model and graphics interface modules

are used in several ways. First, the animation system was used to preview robot operations to ensure that the commanded robot motions did not result in collisions. In addition, the graphics model served as the geometric world model during real-time operations. For example, during manual control of the robot (using remote video cameras), the operator's commands to the robot were evaluated in the graphics environment to ensure no inadvertent collisions would occur. Commanded motions which might have resulted in collisions were automatically interrupted by the control system to ensure safe operation.

REFERENCES

1. Richard Quintero, ed., "DOE/NIST Workshop on Common Architectures for Robotic Systems," NIST Special Publication 784, April 1990.

2. Richard Quintero, ed., "Proceedings of the Second DOE/NIST Workshop on Common Architectures for Robotic Systems," Seattle Washington, 1/23 - 1/24, 1991. (Available from NIST, Robot Systems Division, Building 220, Rm B124, Gaithersburg, MD 20899)

3. W. R. Jaquish, E. H. Shen, and, J Yount "Robotics Technology Demonstration Program for Underground Storage Tank Remediation Program," ANS Winter Meeting, San Francisco, November 10-14, 1991.

4. B. L. Griebenow, L. A. Strope, C. B. Selleck, and J. D. Burke, "Imaging, Inspection, and Characterization System for Underground Storage Tank Remediation," ANS Winter Meeting, San Francisco, November 10-14, 1991.

5. B. K. Christensen, Bl. L. Griebenow, and B. L. Burks," Graphic Model Based Control of Robotic Systems for Waste Remediation," ANS Winter Meeting, San Francisco, November 10-14, 1991.

ACKNOWLEDGEMENTS

END

DATE FILMED
6 / 22 / 93