United States Department of Commerce
National Institute of Standards and Technology

NIST Technical Note 1277

Concept for a Reference Model Architecture for Real-Time Intelligent Control Systems (ARTICS)

James Albus, Richard Quintero, Ronald Lumia,
Martin Herman, Roger Kilmer, and Kenneth Goodwin

NIST Technical Note 1277

Concept for a Reference Model Architecture for Real-Time Intelligent Control Systems (ARTICS)

James Albus
Richard Quintero
Ronald Lumia
Martin Herman
Roger Kilmer
Kenneth Goodwin

Robot Systems Division
Center for Manufacturing Engineering
National Engineering Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899

April 1990

U.S. Department of Commerce
Robert A. Mosbacher, Secretary

National Institute of Standards and Technology
John W. Lyons, Director

National Institute of Standards and Technology
Technical Note 1277
Natl. Inst. Stand. Technol.
Tech. Note 1277
30 pages (Apr. 1990)
CODEN: NTNOEF

U.S. Government Printing Office
Washington: 1990

For sale by the Superintendent of Documents
U.S. Government Printing Office
Washington, DC 20402

Concept for a Reference Model Architecture for Real-Time Intelligent Control Systems (ARTICS)

by

James S. Albus, Richard Quintero, Ronald Lumia,
Martin Herman, Roger Kilmer, and Kenneth Goodwin
Robot Systems Division
National Institute of Standards and Technology

ABSTRACT

This paper presents a concept for the development of a reference model open-system architecture for real-time, sensory interactive, intelligent machine systems. Central to this notion is a desire to accelerate technological development, technology transfer and commercialization of world class control system products in the field of robotics, intelligent machines and automation. A plan is outlined whereby a reference model Architecture for Real-Time Intelligent Control Systems (ARTICS) can be defined through the cooperative efforts of industry, academia and government. As envisioned ARTICS would be a series of evolving guidelines specifying an infrastructure of hardware components, software components, interfaces, communications protocols and application development tools. An ARTICS reference model would make it possible for industry to develop and market a diverse line of control system components which could be interchangeable and realizable on many different vendors' intelligent machine systems platforms.

KEY WORDS:

ARTICS, automation, control systems, intelligent machines, real-time control, reference model control architecture, robotics

TABLE OF CONTENTS

Concept for a Reference Model Architecture for Real-Time Intelligent Control Systems (ARTICS)

by

James S. Albus, Richard Quintero, Ronald Lumia,
Martin Herman, Roger Kilmer, and Kenneth Goodwin
Robot Systems Division
National Institute of Standards and Technology

INTRODUCTION

This paper advocates the development of a reference model open-system Architecture for Real-Time Intelligent Control Systems (ARTICS) as a means to accelerate the pace of technological development in automation and robotics. We believe many of the major bottlenecks in the development of intelligent machine systems could be alleviated, if not eliminated, by the development of a set of ARTICS guidelines.

The pace of commercial and military technological advancement in the fields of robotics, intelligent machine systems and automation is falling short of expectations. Problem complexity is one of the major contributors to this problem. For example, the task of emulating even insect level intelligence, sensory perception, dexterity and functionality has proven to be much more difficult and costly than many had thought. Intelligent robot systems projects typically require bringing together teams of technologists with a broad mix of engineering disciplines and a high level of expertise. Robotics and automation manufacturers must make large investments in both developing custom test-beds and in recruiting and training competent engineering teams in order to compete in this market area. A second problem is the lack of a widely accepted theory, or system architecture that ties together the many disciplines involved in intelligent robot systems. This limits the dissemination of intelligent machine systems technology developed in different parts of the robotics community. This prevents new projects from building upon the foundations laid by previous efforts.

A set of ARTICS guidelines would reduce the impact of problem complexity and would provide an efficient means of transferring technology between projects. Manufacturers will adopt ARTICS guidelines if they believe that their potential profits would be enhanced by an expanded market. This must be driven by traditional market forces (user demand). We need a way to create automation building blocks so that more complex systems can be developed without making the technologies more difficult to understand and to apply and without "reinventing the wheel" each time a new project begins. We believe that a common hardware/software shell structure would facilitate the incremental improvement which would produce rapid advancement in automation and robotics technology.

To summarize the goals advocated by this paper, we suggest that an Architecture for Real-Time Intelligent Control Systems (ARTICS) is needed to:

* reduce the impact of problem complexity in the development of robotic applications

* expand the market for intelligent control system components through open-system interface guidelines and protocols.

* promote portability, inter-operability and modularity of intelligent control system software and hardware

* facilitate technology transfer between intelligent control system projects

* reduce the time, cost, risk, and initial investment required in bringing new, world class, intelligent machine systems and control system products into the market place

2. ARTICS VISION

As envisioned, ARTICS will be a series of guidelines that define a framework of hardware and software system components with well defined interfaces and communications protocols. It would be an evolving series of guidelines, which would be issued in versions and updated as the technology matures.

ARTICS guidelines would specify a reference model infrastructure of hardware components, software components, interfaces, communications protocols, and application development tools. Such a set of guidelines would make it possible for industry to develop and market a diverse line of control system components which could be interchangeable and realizable on many different vendors' control systems platforms.

ARTICS would be designed to facilitate technology and component transfer among the various users and developers, taking advantage of commonalities among otherwise disparate applications such as manufacturing, construction, environmental restoration, mining, space exploration telerobotics, medicine, and military applications of air, land, space, sea-surface, and undersea robotics.

A commercially manufactured ARTICS implementation product would come with libraries of algorithms for planning, task execution, sensor processing and world modeling. These libraries would be user expandable and replaceable. An ARTICS implementation would be fully documented so that users could easily modify or replace any module with a minimum of effort. It would also be commercially maintained, so that users would be able to get help in fixing bugs and making system modifications. In addition, vendors would offer training services to help the user community apply ARTICS products to their applications.

Widely available ARTICS off-the-shelf products would include a target computer system with a backplane and bus configured as a card cage, a local area network to link distributed applications and interface workstations for human/computer interface and software development, a real-time multi-processor/multi-tasking operating system, compilers, debuggers, and CASE tools. ARTICS compliant products could be integrated into an extendable open-system architecture with complete documentation of all hardware and software components.

The ultimate goal would be for ARTICS to evolve into a set of standards for real-time intelligent control systems. Following the example of music stereo systems or microprocessors: the hardware and software of manufacturers world-wide can be interconnected and used in an endless array of applications; and new, innovative component products can be connected as soon as they are introduced into the marketplace. The large market of potential applications will encourage the development of innovative new components, and new innovations will be marketable as soon as they adhere to the standards and interface protocols.

Figure 1 illustrates a possible common system configuration for a Version 1 ARTICS system. It would be organized into three levels.

The top level would consist of a number of workstations on an Ethernet for off-line software development and testing. A number of Computer Aided Software Engineering (CASE) tools, shell programs, simulators, debugging and analysis tools, and compilers for at least C, Ada, and Common LISP would be available.

These workstations might include one or more SUNs, LISP machines, VAXes, Butterflys, Connection Machines, graphics engines and display and image processing machines.

One or more of the workstation machines might also be used for on-line real-time control of processes where response time can exceed 1 second, and in situations where weight, power, and other environmental requirements permit. A real-time, multi-computer, multi-tasking operating system such as real-time UNIX, or MACH would be provided to support this type of operation.

The top level would also support an interface to a gaming environment such as the DARPA SIMNET. This would provide a low cost means for testing and evaluating the performance of intelligent machines in a war gaming environment against manned systems, or other unmanned systems. It would also provide an environment for developing tactics and strategy for using large numbers of intelligent vehicles and weapons systems in large scale battle simulations.

The middle level of the ARTICS system would consist of target hardware, such as single board computers and memory boards of the 680X0 variety, using VME or Multi-bus communications. More than one such bus might be connected via bus gateway cards. This middle level would have a real-time, multi-processor, multi-tasking operating system such as pSOS, VRTX, or MACH capable of supporting response times of ten milliseconds or greater.

The bottom level would consist of special purpose hardware which would interface to the VME or Multi-bus. This level would support high speed parallel processing for images, as well as servo controllers with response times between ten microseconds and ten milliseconds.

ARTICS VERSION 1
COMMON SYSTEM CONFIGURATION

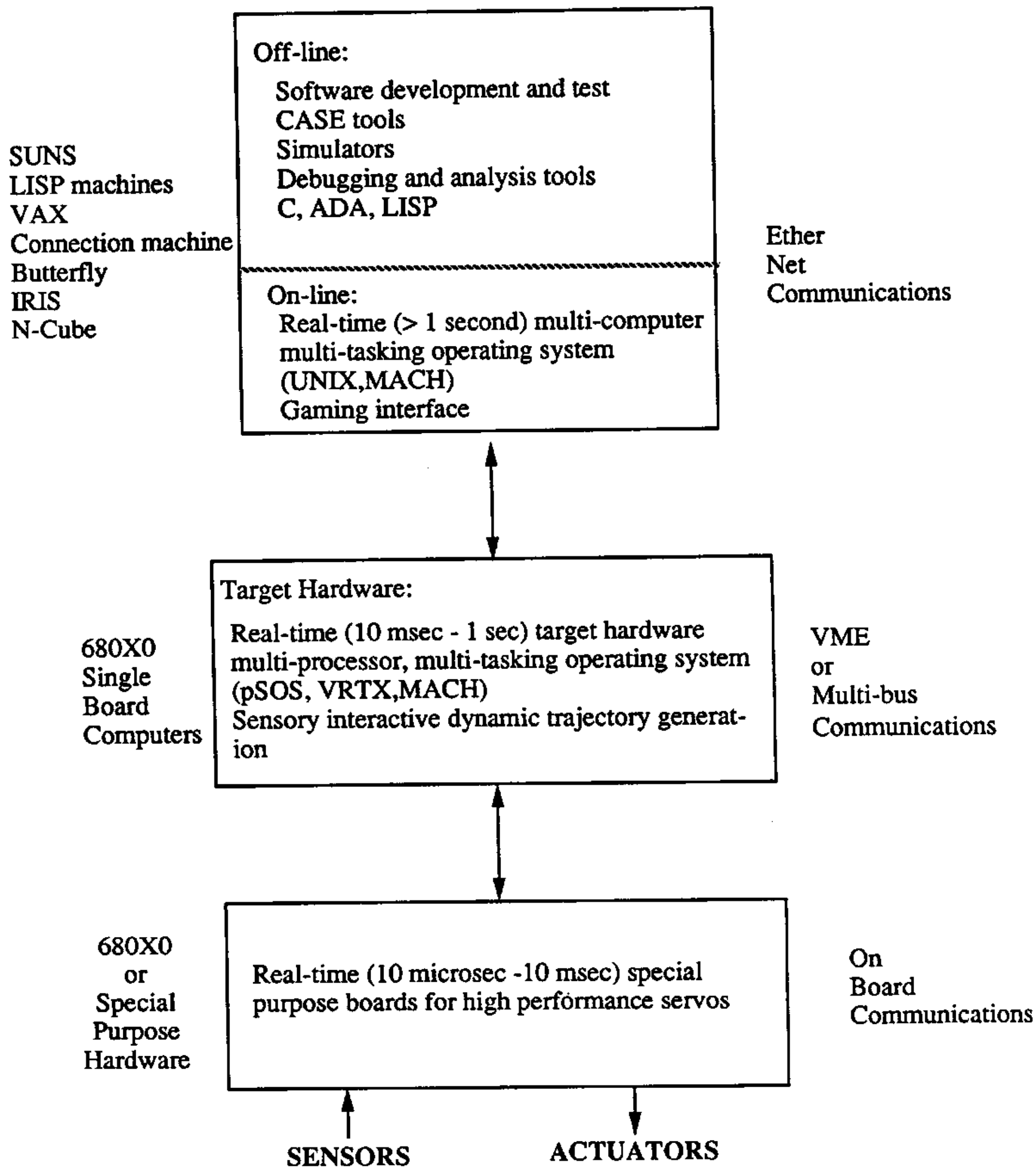

Figure 1

The goal for ARTICS would be an open system architecture with full documentation so that manufacturers and software designers could plug their products into one or more modules with complete compatibility. Modules from one robotic application, such as a remotely piloted air vehicle, could be made convertible to another application, such as a ground vehicle. For example, world model map management and map to egosphere transformation modules might be transferable from one application to another. The hardware, data structure, and interfaces could be common, while the data would be application specific.

Figure 2 shows a possible reference model architecture based the Real-time Control System (RCS) concepts NIST has developed since 1980. These have been implemented in a number of applications including the Automated Manufacturing Research Facility (AMRF), the NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM) [Al 89], the Air Force Next Generation Controller for machine tools and robots, and the control system architecture research conducted for the NIST/DARPA Multiple Autonomous Undersea Vehicle (MAUV) project [Al 88]. The version of ARTICS shown here consists of six hierarchical levels: servo, path dynamics, elemental tasks, individual (vehicle), group (squad), and cell (platoon).

The top (platoon) level of this reference model architecture would have interfaces to a higher (company) level in a battle management system. The bottom (servo) level would interface to actuators and sensors, and operator interfaces would be defined for all levels.

NIST hopes to enlist the cooperation of experts from industry, academia, and government in developing and modifying these concepts into an agreed upon initial set of guidelines. NIST also intends to sponsor research and enlist others to sponsor research, into advanced concepts that will permit the ARTICS guidelines to evolve as technology advances.

REFERENCE ARCHITECTURE
FOR
REAL-TIME INTELLIGENT CONTROL

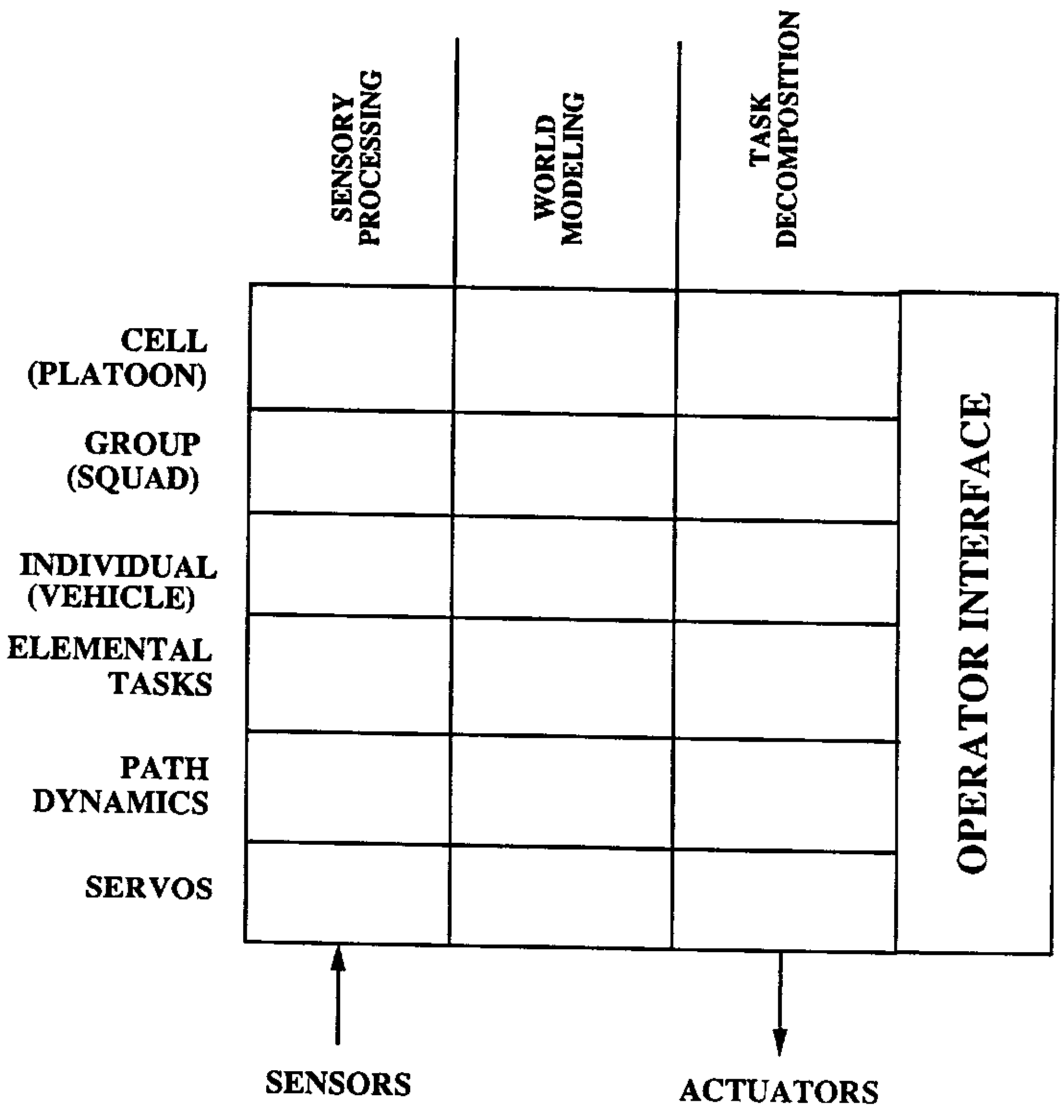

Figure 2

3. REQUIREMENTS

In this section we will begin the process of identifying a set of requirements for an ARTICS architecture. This set of requirements is intended as a "strawman" framework to encourage the robotics community to begin the discourse.

3.1. Extensibility

The goal of extensibility is to provide for both planned and unplanned growth of the application through upgrades of hardware, software or both. Planned growth can mean designing a product line where functionality and cost increase incrementally by building in interfaces and hardware slots for system expansion. Unplanned growth is usually driven by user demand after a product is in the marketplace. Unplanned growth occurs when a product is upgraded to improve performance or functionality by adding to or upgrading the system with new technology which may not have existed or was too expensive when the system was originally designed. The development team should be able to select an appropriate set of modules so as to reach a balance between functionality, user friendliness and cost for their first design with confidence that expanding the design will be feasible. The developers should be able to start with a small inexpensive design and grow the design into a more complex system.

3.1.1. Functional Extensibility

Functional extensibility is defined here as the ability to add new or additional functions to a system or to improve the performance of the system while minimizing and localizing the need for redesign. For example new or improved sensors, actuators communications links or human interface devices might be added which could require a corresponding upgrade to the control system hardware and/or software. Upgrades might also take the form of new or modified functions which a robotic system is to perform without adding new sensors or end-effectors. In a hierarchical design such functions could be added at the same level as (in parallel with) existing functions or as an additional layer. The extensibility requirement demands that the effect of adding functionality to the original design be localized to that part of the design where the function is added (limiting the "ripple-effect" on the design as a whole). This requirement implies that ARTICS should have a modular structure with clearly defined interfaces, command flow and information flow. ARTICS should also allow the development of simple, flat (single layer) structures as well as more complex hierarchical designs.

3.1.2. Temporal Extensibility

Temporal extensibility is defined as the ability to expand the temporal range over which an intelligent robotic system can react to its environment and plan future actions in real-time. For example a real-time intelligent control system might have been originally designed to perform planning functions over a time span of 1 hour and have the ability to react to external events with a response time of tens of milliseconds. Expanding such a system's temporal span of control might involve increasing its planning horizon to several hours and/or increasing its real-time reaction time capability to a millisecond or less. The goal of temporal extensibility is to allow such increased real-time performance while limiting the need for and the extent of redesign.

3.2. Human/Computer Interface Flexibility

ARTICS must provide for a very wide range of human/computer interface. The system designer must be able to select from a large variety of existing and future human interface devices to arrive at an appropriate match between the level of man-in-the-loop complexity and the degree of "user-friendliness" to be achieved in a design. This implies that ARTICS must include very flexible hardware interface guidelines and software protocols. ARTICS must allow the use of human input devices such as: switches, buttons, levers, replica master devices, joy-sticks, track balls, mouse devices, keyboards, microphones, and teach pendants as well as sophisticated devices such as six degree-of-freedom space balls, hand gloves, body gloves, and head position sensors. ARTICS should also accommodate a wide range of human interface display devices such as cathode ray tube (CRT) devices, plasma displays, printers, plotters, color displays, three dimensional displays, analog presentation of data (gages), audio output devices, video displays, helmet mounted displays, as well as devices to stimulate the operator's other senses as appropriate to the application (feel, balance, temperature, vibration, smell, taste, etc.). In short ARTICS must not limit the designer's choice of human interface devices.

3.3. Level of Automation Flexibility

Automation flexibility is defined as the ability to expand or minimize both the degree of automation complexity and the level of man-in-the-loop control. The goal is to allow the robotic system designer to "trade-off" between the functions to be performed by an operator and those which are to be automated. An ARTICS application should be possible as a simple teleoperated or remote control system with the ability to incrementally enhance the level of automation to fully autonomous operation.

As shown in figure 3, a control system application can be thought of as occupying some volume in the space defined by three axes (not necessarily orthogonal). In figure 3 we define the X-axis as representing a high degree of man-in-the-loop interaction at the origin and increasing autonomy to the right. The Y-axis indicates the level of sensory input to the application with increasing sensory capability from the origin (i.e., sensor performance, number and types of sensors). The Z-axis depicts the level of processing and communications power in an application increasing from the origin (including conventional and artificial intelligence hardware and software technologies).

To illustrate the idea of automation flexibility we have placed several examples in the figure. Figure 3 shows that a toggle switch has a high degree of human control, no sensory capability and no processing power. A thermostat is shown as a highly autonomous single sensor device with almost no processing power. A telerobot system has a high degree of human control and also some level of autonomy. A telerobot might also employ moderate levels of sensors, processing and communications power. A pick and place robot can be highly autonomous but it might employ little or no sensory capability and relatively little processing power. Smart weapons systems tend to be highly autonomous and usually include moderately high levels of sensor and processing power. Expert systems applications might require very high levels of processing power. They might also receive and process sensory data and they are typically designed to be highly human interactive with no capability for autonomous control. An intelligent machine is characterized by high levels of processing, communications and sensory power while spanning the X-axis to indicate both high levels of human interactive capability and high levels of autonomous capabilities.

ARTICS must be able to support the development of robotic systems which utilize any combination of automation complexities. That is, the designer must be free to "trade off" functions to be performed as closed-loop autonomous functions (using sensory interactive control) and those system functions to be directed by an operator.

Automation Flexibility

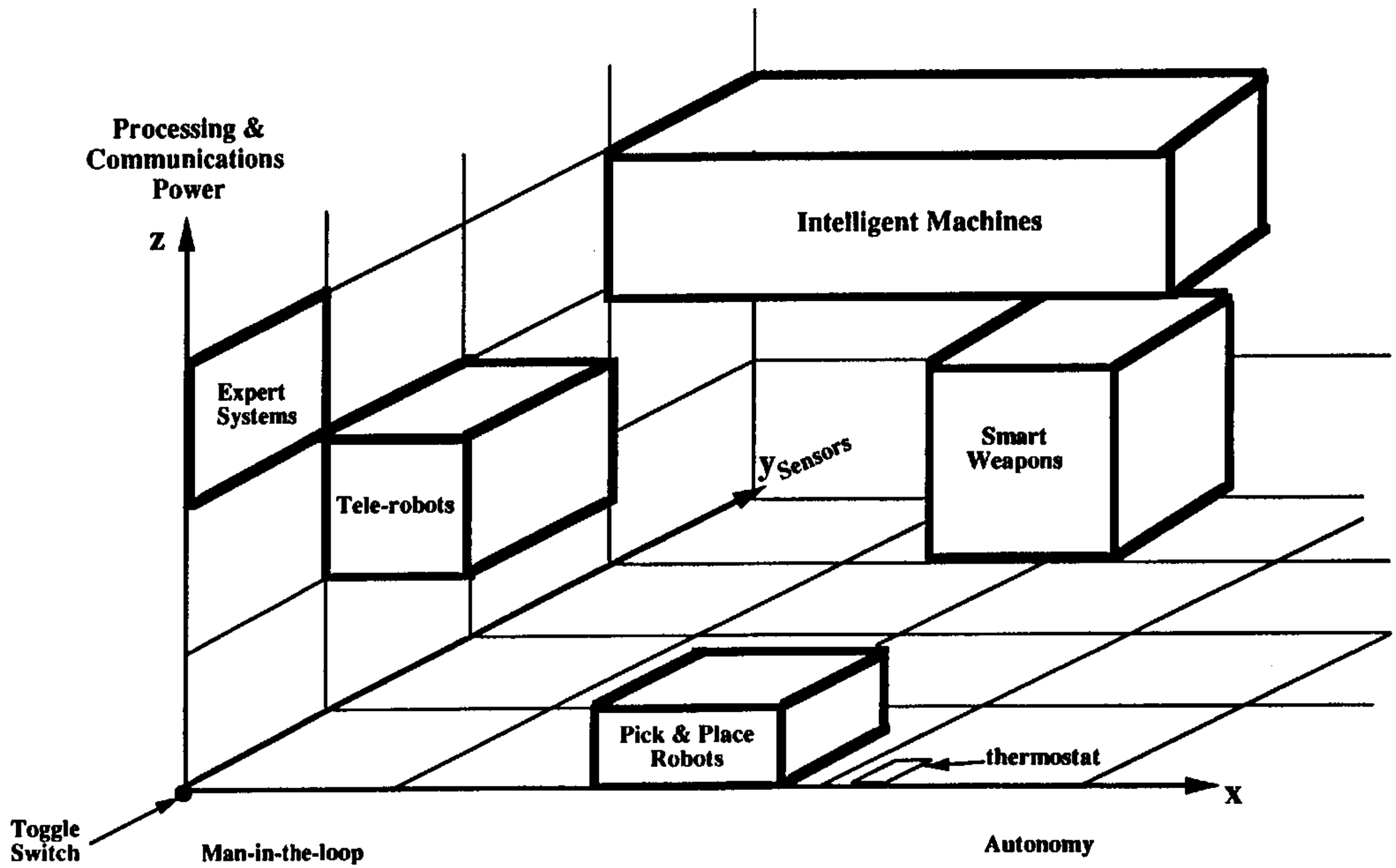

Figure 3

3.3.1. Teleoperation and Remote Control

ARTICS must allow a designer to develop applications employing little or no automation. The designer should be free to develop low-level open-loop control systems where all machine actuator motion is controlled by an operator. A simple example of such a remote control system is a radio controlled vehicle which is controlled via a joy stick by a human operator using human visual contact to close the control loop. When such a system includes sensors the data are normally fed back to the operator for interpretation and action such as a video link between a teleoperator system and the operator. ARTICS should also support the development of teleoperator master/slave systems employing force-reflective feedback to the operator.

3.3.2. Computer Aided Advisory Control

ARTICS must support the development of robotic applications employing computer aided advisory control. Such a system is one in which a computer analyzes the internal system status and the external world state in real-time and advises the human operator as to a recommended course of action. These are man-in-the-loop systems which require positive operator control in order to activate actuators. Such a system might also allow the operator to pose "what if" questions to explore alternative actions based on computer predictions. This requirement suggests the integration of expert systems with teleoperated or manually controlled ("fly by wire") systems.

3.3.3. Traded Control

ARTICS must allow the design of traded-control systems where the operator can selectively choose to employ autonomous subfunctions or retain manual or teleoperator control of machine motions. In a traded control design the operator and the machine alternate in controlling the robotic system.

3.3.4 Shared Control

ARTICS must allow the designer to develop systems employing shared control. For example an end-effector's movement might be controlled autonomously in 2 degrees of freedom (e.g., vertical and x-axis translation) and manually by the operator in a third degree (y-axis translation) of freedom.

3.3.5. Human Override

ARTICS must provide for the development of control systems where an operator can interrupt or override autonomous machine functions while they are in execution. For example a remote control land vehicle might employ an autonomous steering function capable of following a planned driving path and a human override control capability. If we assume the system includes a remote command module for an operator with a real-time video display of the path being traversed by the vehicle and a steering wheel with reflective-force feedback, the operator could employ human override control by simply grasping the steering wheel and manually controlling the path taken by the vehicle. When the operator releases the wheel the system would resume autonomous control.

3.3.6. Human Supervised Control

ARTICS must allow the designer to develop systems with autonomous subfunctions which can be invoked or rejected by a human operator who supervises all machine actions. In this type of implementation low level functions are typically either pre-coded in software by the application developer or learned by the robot in teach mode. These low level autonomous behaviors can then be activated at the operator's discretion.

3.3.7. Autonomous Control

ARTICS must provide the guidelines necessary to implement autonomous control of some or all machine functions. Autonomous control includes simple open-loop pre-coded or learned machine behaviors typically employing position control as well as more complex closed-loop sensory interactive control. Autonomous control also includes complex computerized intelligent machine functions such as sensory processing, situation assessment, world modeling and task decomposition (including real-time planning). ARTICS must be able to accommodate single level intelligent control system architectures or complex multi-level hierarchical control system architectures.

3.3.8. Sensory Interactive Control

ARTICS must be able to support systems which employ sensors to perform closed-loop autonomous control over some or all of their system functions in real-time. Sensors may be used to measure the internal state of the machine in real-time and/or the state of the external environment. For example a torque sensor can be used to constantly measure the forces being exerted by a robot arm joint in order to perform closed-loop autonomous force control of arm movement (compliant motion control) or to provide reflective-force feedback for a teleoperator system. In a more complex control loop a camera can be used to track the movement of an object in a robot's environment in order to coordinate the movement of an end-effector with relation to the object being tracked. Radar, sonar, laser and vision sensor systems can be used to measure the range of objects in the environment and to map their features. The system designer should not be limited in his or her selection of sensors to be applied to a robotic problem. This requirement implies a need for hardware, software and protocol standards for interfacing sensors as well as the opportunity to develop generic software libraries required to acquire, filter, process, model and interpret real-time sensory data.

3.3.9. Mixed Mode Control

ARTICS must be able to accommodate mixed mode control systems designs where the developer is free to design a complex mix of any or all of the modes of control discussed above without violating the ARTICS conceptual architecture precepts.

3.4. Real-Time and Temporal Reasoning

The ARTICS guidelines must provide for real-time control system hardware and software components as well as temporal reasoning software for complex intelligent machine applications. The system designer should have at his or her disposal a real-time operating system, system clock hardware, synchronous and asynchronous software timing control routines and hardware/software interrupt processing components. The design team must be able to select ARTICS compliant modules and interfacing components which will provide a sufficient closed-loop bandwidth to match the real-time constraints of the application. Temporal reasoning software will be needed to keep track of the history of events the intelligent machine has encountered during a task evolution (in order to predict future events) and to reason about time dependent functions (e.g., scheduling tasks, time-out functions, aging facts and data, etc.). Temporal reasoning ability would be a particularly important feature for artificial intelligence products such as expert system shells.

3.5. Distributed System

ARTICS must support intelligent machine applications involving distributed networks of micro-processor and computer control systems as well as stand alone single robot applications. Such applications could include the coordinated control of several robots forming a work cell in a factory environment, several earth moving vehicles forming a coordinated work cell for construction or land reclamation, a number of unmanned military land and air vehicles performing a coordinated mission or a man-machine automated system involving several groups of robots, computer systems and people arranged in a semi-autonomous hierarchy. This requirement suggests a strong need for communications protocols utilizing processor to processor interfaces, local area networks and remote network communications as well as distributed database management guidelines.

3.6. Graceful Degradation

ARTICS should support the development of applications which require graceful degradation. This is particularly important when the application involves military combat or other hazardous environments. This requirement suggests that ARTICS should support the development of control systems made up of many modular components capable of some level of independent function even when other parts of the distributed intelligent machine are no longer functioning.

3.7. Application Independence

An important goal of the ARTICS concept is to achieve a high level of application independence for intelligent machine system components. ARTICS guidelines should make it possible to develop libraries of intelligent machine system software and hardware components which can be easily utilized in a wide variety of applications with little or no modification.

3.7.1. Software Portability

Software guidelines are required to insure that software libraries can be developed which are either directly linkable into new applications (i.e., executable run-time libraries) or portable via a simple process of re-compiling or cross-compiling the software.

3.7.2. Compatibility and Inter-Operability

Hardware and interface guidelines are required to insure that hardware and interface components (card cages, cards, cables, connectors, power systems, cooling systems, network interfaces, peripheral device interfaces, etc.) are compatible and inter-operable even when manufactured by different manufacturers.

3.8. Ease of Use

ARTICS products must be "user friendly". Development environment products and human interface products should make maximum use of menu driven software and on-line documentation. ARTICS components must be well documented. Applications developers should have easy access to product documentation, consultation services and training.

3.9. Cost Effectiveness

ARTICS products should be developed as product lines offering base-line functionality at a cost-effective price and options, upgrades, or more sophisticated products at additional cost. This would allow the application designer to match the control system functionality and user friendliness to the target cost of the control system.

3.10. Development Environment

Users must be supported by a comprehensive set of ARTICS compliant development environment products. Some of the products required would include software development workstations, graphics workstations, micro-processor development systems, local area networks, compilers, cross-compilers, artificial intelligence shells and interpreters, CASE tools, software libraries for sensory processing, world modeling and task decomposition, an ARTICS control system shell program, database management software, real-time operating systems, project management tools, documentation tools, CAD/CAM tools and database translators. Real-time software development tools will also be needed which can calculate execution time budgets for source code modules under development as well as the reserve computing capacity of each CPU being used in the target control system application.

3.11. Simulation and Animation

ARTICS application developers will require a comprehensive set of simulation and animation tools. Guidelines will be required to make simulation and animation tool sets directly compatible with the control system development environment tools. Developers will need the ability to develop applications using simulation tools first and then incrementally substituting target control system components and applications software as they are developed. Powerful debug tools will be required including the ability to run simulations in real-time, slower than real-time and faster than real-time. Tools will also be required to allow a developer to selectively log and display or print simulation events and the data and control variables associated with each event. Even more sophisticated tools such as the DARPA SIMNET will be required in order to develop effective tactics for military combat robotics. Such tools must allow the developer to stage scenarios in order to evaluate the performance of an intelligent control system against human opponents and/or multiple simulated manned or unmanned opponents. Animation tools will be required to allow developers to visualize the results of intelligent control system behaviors as they are being developed in software and before the mechanical development of the machine is completed. Animation tools will also allow off-line development of new control system software after an intelligent machine is in service.

4. APPROACH

To implement the ARTICS concept a consensus must be achieved in several areas of the common control system architecture. Ideally there would be early agreement upon a conceptual framework for the development of intelligent machine control systems. Such a conceptual framework would provide developers with a common design philosophy to guide the development of new robotic applications and control system products. There are a number of researchers currently working in the field of control architectures and certainly this work should not only be continued but it should be pursued even more aggressively. A number of control architectures should be considered and evaluated against some set of agreed upon common control system requirements and finally a common conceptual architecture must be derived from the results of the process. More than likely such an architecture would include the ideas of a number of researchers as well as strong input from the user community. The following is a list of recent research in this area:

- Action Networks [Ni 89]
- Autonomous Land Vehicle (ALV) [Lo 86]
- Automated Manufacturing Research Facility (AMRF) [Si 83, Al 81]
- Control in Operational Space of a Manipulator-with-Obstacles System (COSMOS) [Kh 87]
- COmmunications Database with GEometric Reasoning (CODGER) [Sh 86]
- Field Materiel-Handling Robot (FMR) [Mc 86]
- Generic Vehicle Autonomy (GVA) [Gr 88]
- Hearsay II [Le 75]
- Hierarchical Control [Sk 89, Sk 87, Ko 88, Ko 88]
- Hierarchical Real-time Control System (RCS) [Ba 84, Al 81]

- Intelligent Control [Sa 85]
- Intelligent Task Automation (ITA) [Bl 88]
- Manufacturing Automation System/Controller (MAS/C) [Ho 88]
- Multiple Autonomous Undersea Vehicles (MAUV) [Al 88]
- NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM) [Al 89]
- Pilot's Associate [Sm 87]
- Robot Control "C" Library (RCCL) [Ha 86]
- Robot Schemas [Ly 89]
- Soar: Architecture for General Intelligence [La 86]
- Subsumption Architecture [Br 86]
- Task Control Architecture (TCA) [Si 89]
- Tech-based Enhancement for Autonomous Machines (TEAM) [Sz 88]
- University of New Hampshire (UNH) Time Hierarchical Architecture [Ja 88]

The first version of ARTICS should incorporate generic real-time control systems concepts developed in a number of industry, academic, DARPA, Air Force, Navy, Army, and NIST research programs, including the following projects: Intelligent Task Automation, Autonomous Land Vehicle, Pilot's Associate, Multiple Autonomous Undersea Vehicles, NASA space station Flight Telerobotic Servicer, Field Materiel-Handling Robot, TEAM, and Automated Manufacturing Research Facility.

The National Institute of Standards and Technology (NIST) has been conducting collaborative research with industry, academic and other government partners in the generic concepts of hierarchical real-time control for a number of years. This work has resulted in the development of the NIST Hierarchical Real-Time Control System (RCS) which we offer as a candidate conceptual architecture for consideration by an ARTICS guideline development body [Al 89, Al 88, Ba 84]. RCS has been under development in the NIST Automated Manufacturing Research Facility (AMRF) [Si 83, Al 81] since 1980 and in various mobile intelligent robotics projects sponsored by DoD agencies and NASA since 1985. We believe RCS is an excellent "strawman" to serve as a catalyst for beginning the ARTICS concept dialogue.

Consensus must also be reached on several other common architecture components. These include the specifications of control systems hardware components such as: controller boards, backplanes, connectors, cables, human interface devices, power systems, sensors, other peripheral devices and communications networks. Software component specifications will be needed for: real-time operating systems, data-base management systems, artificial intelligence technologies, software languages, communications software and software libraries. In addition specifications for communications, timing synchronization and control between modules at all levels of the system in the form of interface protocols are required for: inter-process communication, backplane communication, local area network communication, communication with humans and remote network communications.

There are a number of government efforts under way that should be factored into the process of defining an initial set of common architecture components. Some of these include:

- The NIST Federal Information Processing Standard (FIPS)
- The NIST Government Open Systems Interconnection Profile (GOSIP)
- The Navy's Next Generation Computer Resources (NGCR) program
- The Air Force's Next Generation Controller (NGC) program
- The Army's Standard Army Vetronic Architecture (SAVA) program

In addition there is a Department of Energy interest in establishing guidelines for robotic systems needed in their Environmental Restoration and Waste Management Program, the U.S. Bureau of Mines Pittsburgh Research Center is conducting research in automation systems for coal mining and there are a number of DARPA programs (past, present and on-going) which are producing relevant technologies.

In Appendix A we have attempted to compile a "strawman" list of potential ARTICS components. This list is intended to serve as a catalyst or a point of departure to begin the ARTICS dialogue. It is not our intent to dictate a predetermined set of arbitrary standards. The reader is invited to review Appendix A and to offer his or her own insights as to the proper make-up of a family of ARTICS components.

5. REFERENCE MODEL DEVELOPMENT PLAN

5.1. Guideline Evolution Process

ARTICS must be able to evolve as technological progress is made. It will be important to create an organizational structure that can coordinate the process of evaluating change and update proposals and a process for achieving consensus on the release of new versions of the ARTICS guidelines. Such an organization will need a steering committee made up of leading experts in the field of robotics, intelligent machines and automation from industry, academia and government.

5.2. Research Support

A strong well funded research program will be needed to support the development and evolution of ARTICS guidelines. Such a program could be coordinated and administered through NIST with funding support from either a voluntary organization such as an industry consortium, through government sponsorship or both. The program administrator would use the funding to support grants and research contracts to industry, academia and government laboratories in order to develop and demonstrate innovative new approaches to real-time intelligent machine control systems technology which could lead to marketable improvements in ARTICS compliant products.

5.3. Coordination Structure

An ARTICS development effort could take the form of a voluntary organization much like the Initial Graphics Exchange Specification (IGES)/Product Data Exchange Specification (PDES) organization [Ig 89] chaired by NIST. Alternatively the reference model could be developed by a major user of the technology such as the Department of Defense in the form of a military specification (MILSPEC). In either case working groups will be needed to steer the ARTICS development and to document and distribute the results. Once an initial set of ARTICS guidelines has been agreed to it can be submitted to one or more national or international standards organizations as a proposed standard (e.g., ANSI, EIA, IEC, IEEE, ISO, RIA, etc.).

5.4.. Testing and Validation

An ARTICS testbed facility will be needed to serve as a focal point for research and testing. The facility would provide researchers and users a place where new ideas can be pursued, where bench mark tests can be developed and where product validation can be performed to certify compliance with the established ARTICS reference model. The NIST Robot Systems Division is planning to develop and maintain a facility for intelligent machine systems research beginning in 1991 as a national resource. We believe this facility will be ideally suited to also serve as a ARTICS testbed.

5.5. Community Input

The NIST Robot Systems Division conducted a limited inquiry as a two-phase Delphi on February 10, 1989 and May 30, 1989 [Ro 89] with the following summary results:

Respondents to the Standard Architecture for Real-time Intelligent Control System (SARTICS) Enquiry:

	Phase I	Phase II
Number Responding	35	23
Government	17	13
Industry	12	6
Academia	6	4
Ave years experience	9	7

Response:

Is there a need for SARTICS ?			
Phase I	32 yes	1 no	2 don't know
Phase II	19 yes	1 no	2 don't know

Do you wish to be involved in the development of SARTICS ?			
Phase I	25 yes	1 no	9 don't know
Phase II	16 yes	2 no	5 don't know

How would you rate the importance of SARTICS ?
Scale: 0 = not important
5 = very important

Phase I Moderately to Very Important (mean 3.6)
Phase II Moderately to Very Important (mean 3.5)

Respondent application areas:
Air, land, sea, space, manufacturing, nuclear, mining

This preliminary survey appears to indicate a considerable interest in the ARTICS development concept. We believe the next step is to hold a series of workshops on ARTICS to provide a forum for community discussion and to test the potential for community participation in an ARTICS development effort.

6. CLOSING COMMENTS

Readers of this paper are invited to contact the authors with comments, criticisms, insights or suggestions concerning the ARTICS concept presented here. We would also be interested in your ideas with regard to establishing mechanisms to implement ARTICS and to organize an ARTICS reference model development body. Correspondence should be addressed as follows:

National Institute of Standards and Technology
Dr. James S. Albus
Chief, Robot Systems Division
Center for Manufacturing Engineering
Bldg. 220, Room B124
Gaithersburg, MD 20899

301-975-3418 FAX# 301-990-9688 FTS# 879-3418

7. REFERENCES

[Al 89] J.S. Albus, H.G. McCain, and R. Lumia, "NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)," NIST (formerly NBS) Technical Note 1235, April 1989 Edition.

[Al 88] J.S. Albus, "System Description and Design Architecture for Multiple Autonomous Undersea Vehicles Project," NIST Technical Note 1251, September 1988.

[Al 81] J.S. Albus, "Brains, Behavior and Robotics," BYTE/McGraw-Hill, Petersborough, NH, 1981.

[Ba 84] A.J. Barbera, J.S. Albus, M.L. Fitzgerald, and L.S. Haynes, "RCS: The NBS Real-Time Control System," Robots 8 Conference and Exposition, Detroit, MI, June 1984.

[Bl 88] R. Blair, "Intelligent Task Automation Interim Technical Report," Report No. II-6, Honeywell Corporate Systems, Golden Valley, MN, August 1988.

[Br 86] R.A. Brooks, "A Robust Layered Control System For A Mobile Robot," IEEE Journal of Robotics and Automation, Vol. RA-2, No. 3, March 1986.

[Gr 88] M.D. Grover, et. al., "An Autonomous Undersea Vehicle Software Testbed," Proceedings of the Fifteenth Annual Technical Symposium of the Association for Unmanned Vehicle Systems, San Diego, CA, June 6-8, 1988.

[Ha 86] V. Hayward, R.P. Hayward, "Robot Manipulator Control Under Unix RCCL: A Robot Control "C" Library," International Journal of Robotics Research, Vol. 5, No. 4, Winter 1986.

[Ho 88] Honeywell Inc., "MAS/C System Overview," MA70-100, Release C, Honeywell Inc., Phoenix, AZ, May 1988.

[Ig 89] "Welcome to IGES/PDES, Newcomer Material," July 1989, available through the Chairman, IGES/PDES, NIST, Bldg. 220, Room A150, Gaithersburg, MD, 20899.

[Ja 88] J. Jalbert, "EAVE III Untethered AUV Submersible," Proceedings of the Fifteenth Annual Technical Symposium of the Association for Unmanned Vehicle Systems, San Diego, CA, June 6-8, 1988.

[Kh 87] O. Khatib, "A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation," IEEE Journal of Robotics and Automation, Vol. RA-3, No. 1, February 1987.

[Ko 88] W. Kohn, T. Skillman, "Hierarchical Control for Autonomous Space Robots," Proceedings of the AIAA Guidance, Navigation, and Control Conference, Minneapolis, Minnesota, Aug 15-17, 1988.

[Ko 88] W. Kohn, "Declarative Theory for Relational Controllers," Proceedings of the IEEE Control Decision Conference, Austin, Texas, Dec 7-9, 1988.

[La 86] J. E. Laird, A. Newell, P. S. Rosenbloom, "Soar: An Architecture for General Intelligence," Carnegie-Mellon University paper CMU-CS-86-171, December 1986.

[Le 75] V.R. Lesser, et. al. "Organization of the Hearsay II Speech Understanding System," IEEE Trans. on ASSP, Vol. 21, No.1, 1975, p. 11.

[Ly 89] D. M. Lyons, M. A. Arbib, "A Formal Model of Computation for Sensory-Based Robotics," IEEE Transactions on Robotics and Automation, Vol. 5, No. 3, June 1989.

[Lo 86] J. Lowerie, et. al. "Autonomous Land Vehicle," ETL- 0413 Martin Marietta Denver Aerospace, Denver, Co, July 1986.

[Mc 86] H. G. McCain, et. al. "A Hierarchically Controlled Autonomous Robot for Heavy Payload Military Field Applications," Proceedings of the International Conference on Intelligent Autonomous Systems, Amsterdam, The Netherlands, December 8-11,1986.

[Ni 89] N. Nilsson, "Action Networks," Proceedings from the Rochester Planning Workshop: From Formal Systems to Practical Systems, J. Tenenberg, et al. (eds.), University of Rochester, New York, 1989.

[Ro 89] Robotic Technology Incorporated, "Standard Architecture for Real-Time Control Systems (SARTICS), Final Report," NIST Contract No. 43NANB909769, dated 21 Nov. 1988, Submitted to NIST Robot Systems Division, Gaithersburg, MD, 3 July 1989.

[Sa 85] G.N. Saridis, "Foundations of the Theory of Intelligent Controls," IEEE Workshop on Intelligent Control, Troy, August 1985, p. 23.

[Sh 86] S.A. Shafer, et. al. "An Architecture for Sensor Fusion in a Mobile Robot," Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, CA, April 7-10, 1986, pp. 2002-2010.

[Sk 89] T. L. Skillman, K. Hopping, "Dynamic Composition and Execution of Behaviors in a Hierarchical Control System," SPIE Mobile Robotics IV, Philadelphia, Nov 7- 8, 1989.

[Sk 87] T. L. Skillman, W. Kohn, et al."Blackboard Based Hierarchical Intelligent Control System," Proceedings of the AIAA Conference on Computers in Aerospace VI, Wakefield, MA, Oct 7-9, 1987.

[Si 89] R. Simmons, T. Mitchell, "A Task Control Architecture for Mobile Robots," Computer Sciences Dept., Carnegie Mellon Univ., Pittsburgh, PA, AAAI '89 Spring Symposium, Robotic Navigation, 1989.

[Si 83] J.A. Simpson, R.J. Hocken, and J.S. Albus, "The Automated Manufacturing Research Facility of the National Bureau of Standards," Journal of Manufacturing Systems, Vol.1, No. 1, pg. 17, 1983.

[Sm 87] D. Smith, M. Broadwell, "Plan Coordination in Support of Expert Systems Integration", Proceedings of the Knowledge-Based Planning Workshop, Austin, TX, December 8-10, 1987.

[Sz 88] S. Szabo, H. A. Scott, R. D. Kilmer, "Control System Architecture for the TEAM Program," Proceedings of the Second International Symposium on Robotics and Manufacturing Research, Education and Applications, Albuquerque, NM, November 16-18, 1988.

APPENDIX A.

LIST OF POTENTIAL REFERENCE MODEL COMPONENTS

An initial set of potential reference model components is presented here to elicit comment and begin the dialogue, not to mandate a predetermined set of arbitrary standards. The suggested components might include the following:

- Target Controller Hardware
  - a standard embedded system backplane and bus architecture (e.g., VME, Multi-bus, NuBus)
  - a set of standard computer bus compatibility options
  - a standard set of plug-in computer boards
    - . high, medium and standard speed CPUs (e.g., Motorola 68020, 68030, Intel 8086, 286, 386)
    - . low power CPUs (i.e., CMOS)
    - .CPU for symbolic processing
    - . computer RAM memory boards and low power RAM (e.g., CMOS)
    - . networking boards (i.e., Ethernet)
    - . I/O communications boards (RS-232, IEEE 488, printer I/F, plotter I/F, etc.)
    - . image warping processor board
    - . synchro to digital, resolver to digital, D/S and D/R converters
  - a set of mass storage peripherals (e.g., Winchester disc, optical disc, bubble memory, tape, etc.)
  - a set of general purpose I/O devices
  - a set of standard connectors and cables
  - a very low power, standby controller board
  - a self powered time standard (e.g., a battery powered clock board)
  - a watchdog timer
  - power supplies, cooling devices, etc.
  - a set of data recorders (e.g., VHS video recorder, audio recorder, etc.)
  - two-way communications devices (e.g., RF, acoustic, microwave, satellite, etc.)

- Target Controller Run-Time Software
  - a hierarchical real-time control system shell program
  - a standard real-time operating system (e.g., Condor, pSOS, VRTX)
  - a standard distributed object oriented database management system
  - an extendible run-time library of software modules for:
    - . path planning
    - . obstacle avoidance
    - . navigation
    - . piloting
    - . object representation and recognition
    - . map representation and multi-level topography (i.e., on-land, underwater, under ice, underground, etc.)
    - . task representation and scheduling

. resource representation and management
. two-way remote communications
. remote command and control/teleoperation
. image processing
. acoustic signal processing
. tactile sensor processing
. proximity data processing
. range data processing
. temperature sensing
. altitude and depth data
. torque data
. force data

- Data Formats and Interface Protocol Definitions for Handling General Purpose Sensors
  - position sensors (INS, GPS, Loran, Omega, etc.)
  - attitude sensors (roll, pitch, yaw, etc.)
  - direction sensors (compass)
  - speed sensors
  - acceleration sensors
  - gyro stabilized video cameras
  - fully automatic still camera
  - acoustic sensors
  - range sensors (laser, radar, sonar, etc.)
  - altitude and depth sensors
  - tactile sensors
  - proximity sensors
  - temperature sensors
  - pressure sensors
  - force sensors
  - torque sensors
  - stress sensors
  - radiation sensors
  - environmental medium sensors (air, water, space, etc.)

- Data Formats and Interface Protocol Definitions for Handling General Purpose Human Interface Devices used for Monitoring, Command and Control
  - alphanumeric displays (e.g., CRTs)
  - graphics displays
  - video displays
  - audio reproduction
  - voice command interpreter
  - mouse, joy-stick, ball tabs, light-pens, keyboards, etc.
  - tactile, vibration, acceleration simulator interface
  - head mounted 3D displays, foveal/peripheral displays, heads-up displays, eye tracking systems

- Software Development Environment
  - an Ethernet network of computer workstations with interfaces for: SUN, MicroVAX, MacIntosh, PCs, NeXT, LISP machines, Butterfly, WARP, Connection Machine, IRIS, E&S, etc.
  - symbolic processing workstations (e.g., Prolog)
  - high speed 3D graphics workstations
  - a shared mass storage system / network server
  - a set of shared general purpose peripherals (e.g., laser printers, plotters, flying spot scanner, etc.)
  - a software development system for the target system
  - a development environment operating system (e.g., Mach, UNIX)
  - CASE tools (e.g., CARDtools, STATEMATE, TEAMWORK, etc.)
  - task decomposition tools to aid in defining task precedence, task durations, concurrency, resource requirements and task data exchange formats
  - a set of compiler languages (e.g., ADA, C++, FORTRAN etc.)
  - a set of symbolic processing shell programs and languages (Common LISP, PROLOG, etc.)
  - a standard object oriented database management system
  - a standard map and topological information database management system
  - target system emulators and cross compilers
  - software debugging tools
  - a project management tool
  - a document processing tool
  - IGES translator
  - PDES translator
  - CAD/CAM tools (e.g., PADL, GEOMOD, CV, ALPHA2, SILMA, CADDAM, CATIA, AUTOCAD, etc.)

- Simulation
  - SIMNET simulation interface
  - laboratory simulation software tools (3D animation)

NIST-114A
(REV. 3-89)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NIST/TN-1277

2. PERFORMING ORGANIZATION REPORT NUMBER

3. PUBLICATION DATE
April 1990

4. TITLE AND SUBTITLE
Concept for a Reference Model Architecture for Real-Time Intelligent Control Systems (ARTICS)

5. AUTHOR(S)
James Albus, Richard Quintero, Ronald Lumia, Martin Herman, Roger Kilmer, Kenneth Goodwin

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)
U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
GAITHERSBURG, MD 20899

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED
Final

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

Same as item #6

10. SUPPLEMENTARY NOTES

☐ DOCUMENT DESCRIBES A COMPUTER PROGRAM; SF-185, FIPS SOFTWARE SUMMARY, IS ATTACHED.

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

This paper presents a concept for the development of a reference model open-system architecture for real-time, sensory interactive, intelligent machine systems. Central to this notion is a desire to accelerate technological development, technology transfer and commercialization of a world class control system products in the field of robotics, intelligent machines and automation. A plan is outlined whereby a reference model Architecture for Real-Time Intelli - gent Control Systems (ARTICS) can be defined through the cooperative efforts of industry, academia and government. As envisioned ARTICS would be a series of evolving guidelines specifying an infrastructure of hardware components, software components, interfaces, communications protocols and applications development tools. An ARTICS reference model would make it possible for industry to develop and market a diverse line of control system components which could be interchangeable and realizable on many different vendors' intelligent machine systems platforms.

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)
ARTICS, automation, control systems, intelligent machines, real-time control, reference model control architecture, robotics

13. AVAILABILITY

- [X] UNLIMITED
- [ ] FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).
- [X] ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.
- [ ] ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES
30

15. PRICE

ELECTRONIC FORM

NIST Technical Publications

Periodical

Journal of Research of the National Institute of Standards and Technology—Reports NIST research and development in those disciplines of the physical and engineering sciences in which the Institute is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Institute's technical and scientific programs. Issued six times a year.

Nonperiodicals

Monographs—Major contributions to the technical literature on various subjects related to the Institute's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NIST, NIST annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NIST under the authority of the National Standard Data Act (Public Law 90-396). NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published quarterly for NIST by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements are available from ACS, 1155 Sixteenth St., NW., Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Institute on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NIST under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NIST administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NIST research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order the **above** *NIST publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.*

Order the **following** *NIST publications—FIPS and NISTIRs—from the National Technical Information Service, Springfield, VA 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NIST pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NIST Interagency Reports (NISTIR)—A special series of interim or final reports on work performed by NIST for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service, Springfield, VA 22161, in paper copy or microfiche form.

U.S. Department of Commerce
National Institute of Standards and Technology
(formerly National Bureau of Standards)
Gaithersburg, MD 20899

Official Business
Penalty for Private Use $300