A Canonical Architecture for Intelligent Machine Systems

James S. Albus
Robot Systems Division
Manufacturing Engineering Laboratory
National Institute of Standards and Technology

Introduction

The study of intelligent machine systems is an extremely active field. Many millions of dollars per year are now being spent in Europe, Japan, and the United States on computer integrated manufacturing, robotics, and intelligent machines for a wide variety of military and commercial applications. Research in learning automata, neural nets, and brain modeling has given insight into learning and the similarities and differences between neuronal and electronic computing processes. Computer science and artificial intelligence is probing the nature of language and image understanding, and has made significant progress in rule based reasoning, planning, and problem solving. Game theory and operations research have developed methods for decision making in the face of uncertainty. Robotics and autonomous vehicle research has produced advances in real-time sensory processing, world modeling, navigation, trajectory generation, and obstacle avoidance. Research in automated manufacturing and process control has produced intelligent hierarchical controls, distributed databases, representations of object geometry and material properties, data driven task sequencing, network communications, and multiprocessor operating systems. Modern control theory has developed precise understanding of stability, adaptability, and controllability under various conditions of feedback and noise. Research in sonar, radar, and optical signal processing has developed methods for fusing sensory input from multiple sources, and assessing the believability of noisy data.

Progress is rapid, and there exists an enormous and rapidly growing literature in each of the areas mentioned above. What is lacking is a general theoretical model, or canonical architecture of intelligent systems, which ties all these separate fields of knowledge into a unified framework. This paper suggests such a framework.

The Elements of Intelligence

The elements of intelligence and their relationship to each other are illustrated in Figure 1.

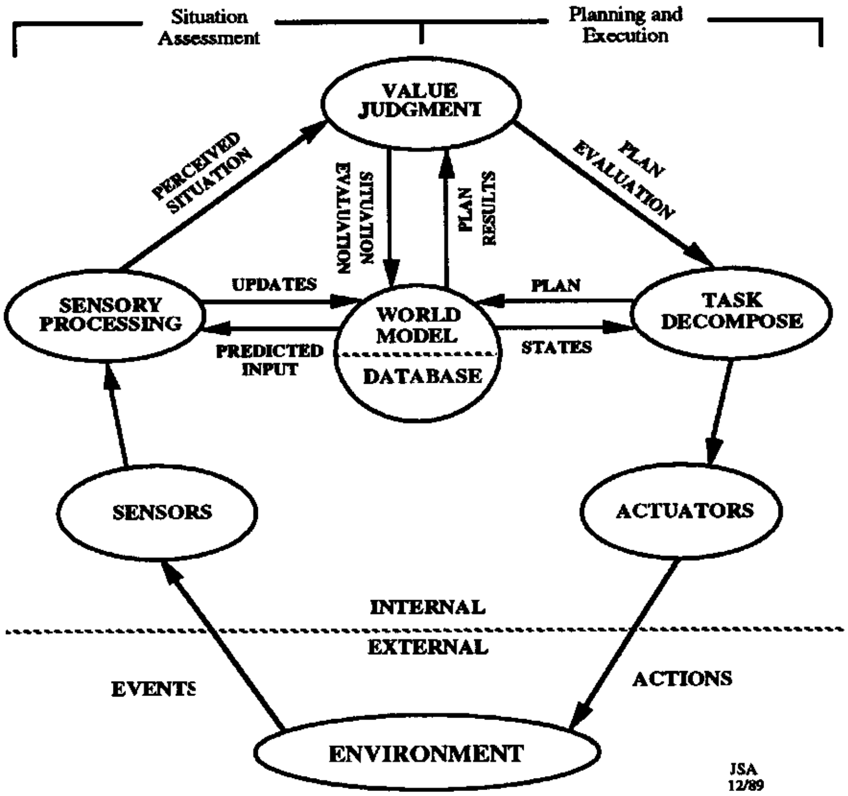


Figure 1. The elements of intelligence and the functional relationships between them.

a. ACTUATORS -- Output from an intelligent system derives from actuators which move, exert forces, and position arms, legs, hands, and eyes. Actuators generate forces to point sensors, excite transducers, move manipulators, handle tools, steer and propel locomotion. An intelligent system may have tens, hundreds, or even thousands of actuators, all of which must be coordinated in order to perform tasks and accomplish goals. Natural actuators are muscles and glands. Machine actuators are motors, pistons, valves, solenoids, and transducers.

b. SENSORS -- Input to an intelligent system derives from sensors. These may include visual brightness and color sensors; tactile, force, torque, position detectors; velocity, vibration, acoustic, range, smell, taste, pressure, and temperature measuring devices. Sensors may be used to monitor both the state of the external world and the internal state of the intelligent system itself. Sensors provide input to a sensory processing system.

c. SENSORY PROCESSING -- Perception takes place in a sensory processing system that compares observations with expectations generated by an internal world model. Sensory processing algorithms integrate similarities and differences between observations and expectations, over time and space, so as to detect events, and recognize features, objects, and relationships in the world. Sensory input data from a wide variety of sensors over extended periods of time may be fused into a consistent unified perception of the state of the world. Sensory processing algorithms compute distance, shape, orientation, surface characteristics, physical and dynamical attributes of objects and regions of space. Sensory processing may include recognition of acoustic signatures, speech, and interpretation of language.

d. WORLD MODEL -- The world model is the intelligent system's best estimate of the state of the world. The world model includes a database of knowledge about the world, plus a database management system that stores and retrieves information. The world model also contains a simulation capability which generates expectations and predictions. The world model thus can provide answers to requests for information about the present, past, and probable future states of the world. The world model provides this information service to the task decomposition system, so that it can make intelligent plans and behavioral choices, and to the sensory processing system, in order for it to perform correlation, model matching, and model based recognition of states, objects, and events. The world model is kept up-to-date by the sensory processing system.

e. VALUES -- The value system makes value judgments as to what is good and bad, rewarding and punishing, important and trivial. The value system evaluates both the observed state of the world and the predicted results of hypothesized plans. It computes costs, risks, and benefits both of observed situations and of planned activities. The value system thus provides the basis for choosing one action as opposed to another, or for acting on one object as opposed to another. The value system also computes the probability of correctness and assigns believability and uncertainty parameters to world model state estimations.

f. TASK DECOMPOSITION -- Behavior is generated in a task decomposition system that plans and executes tasks by decomposing them into subtasks, and by sequencing these subtasks so as to achieve goals. Goals are selected and plans generated by a looping interaction between task decomposition, world modeling, and value judgment functions. The task decomposition system hypothesizes plans, the world model predicts the results of those plans, and the value judgment system evaluates those results. The task decomposition system then selects the plans with the best evaluations for execution. Task decomposition monitors the execution of task plans, and modifies existing plans whenever the situation requires.

In many cases, intelligent task decomposition requires the ability to reason about space and time, geometry and dynamics, and to formulate or select plans based on values such as cost, risk, utility, and goal priorities. Task planning and execution often must be done in the presence of uncertain, incomplete, and sometimes incorrect information.

In order for task decomposition to succeed in a dynamic and unpredictable world, it must be accomplished in real-time. In order to achieve real-time task decomposition, it is necessary to partition the planning problem into a hierarchy of levels with different temporal planning horizons and different degrees of detail at each hierarchical level. Once this is done, it is possible to employ a multiplicity of planners to simultaneously generate and coordinate plans for many different subsystems at many different levels.

The System Architecture of Intelligence

Each of the elements of intelligent systems are reasonably well understood. The phenomena of intelligence, however, requires more than a set of disconnected elements. Intelligence requires an interconnecting system architecture that enables the various system components to interact and communicate with each other in intimate and sophisticated ways.

A system architecture is what partitions the elements of intelligence into computational modules, and interconnects the modules in networks and hierarchies. It is what enables the task decomposition system to direct sensors, and to focus sensory processing algorithms on objects and events worthy of attention, ignoring things that are not important to current goals and task priorities. It is what enables the world model to answer queries from task decomposition modules, and make predictions and receive updates from sensory processing modules. It is what communicates the value state-variables that characterize the success of behavior and the desirability of states of the world.

A number of intelligent system architectures have been proposed, and a few have been implemented. [1-10] The model of intelligence that will be discussed here is largely based on the Real-time Control System (RCS) that has been implemented in a number of versions over the past 13 years at the National Institute for Standards and Technology (NIST formerly NBS).

The Real-time Control System (RCS)

The Real-time Control System RCS was first implemented by Barbera for laboratory robotics in the mid 1970's [11]. It was based on the concept of a hierarchy of CMAC(Cerebellar Model Arithmetic Computer) [12] controllers developed earlier by Albus. Each CMAC could implement a function of the form

$$\mathbf{C}_{i-1} = \mathbf{H}\,(\mathbf{C}_i, \mathbf{F}_i)$$

where $\mathbf{C}_i$ is the command input to level(i)

$\mathbf{F}_i$ is the feedback input to level(i)

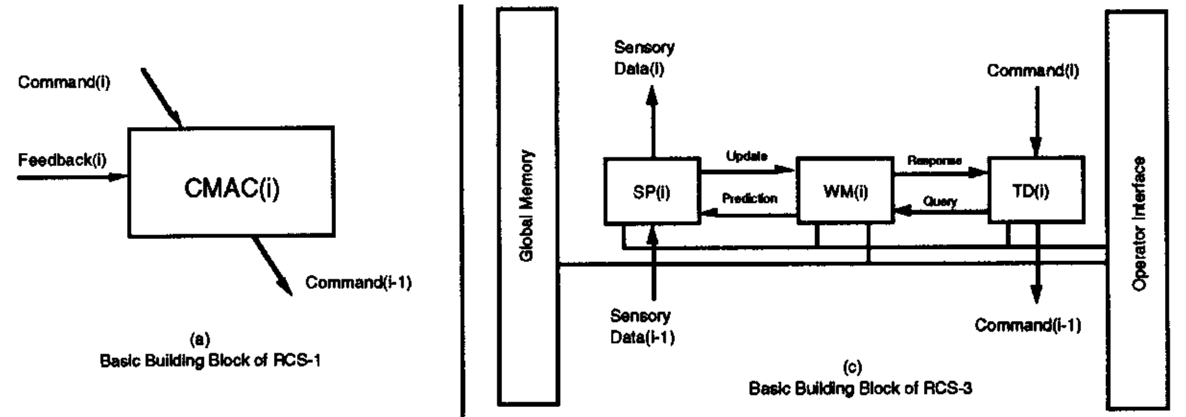
$\mathbf{C}_{i-1}$ is the command output from level(i) to level(i-1)

$\mathbf{H}$ is a smooth single-valued multi-variable function

The basic building block of RCS-1, shown in Figure 2(a), was used to construct a hierarchy of control levels. At each level, the CMAC H function was implemented by a finite state machine defined by a state-table.

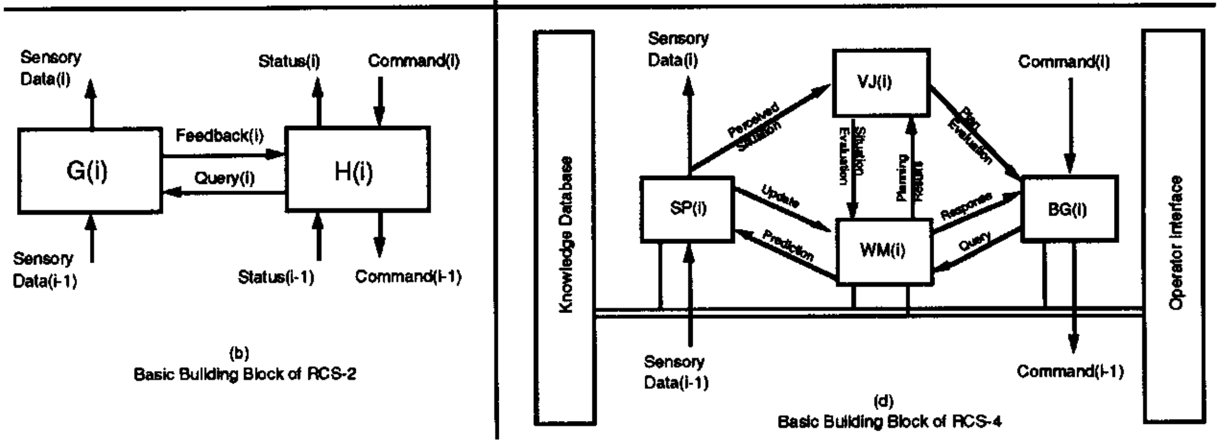
The next generation, RCS-2, was developed by Barbera, Fitzgerald, and others [7] for manufacturing control in the NIST Automated Manufacturing Research Facility (AMRF) during the early 1980's [13-19]. The basic building block of RCS-2 is shown in Figure 2(b). The H function consisted of a state-table executor. The G function consisted of a number of image processing algorithms. RCS-2 was used to define an eight level hierarchy consisting of Servo, Coordinate Transform, E-Move, Task, Workstation, Cell, Shop, and Facility levels of control. Only the first six levels were actually built. Two of the AMRF workstations fully implemented five levels of RCS-2. The control system for the Army Field Material Handling Robot (FMR)[20] was also implemented in RCS-2, as were the Army TMAP and TEAM semi-autonomous land vehicle projects [21].

RCS-3 was designed for the NBS/DARPA Multiple Autonomous Undersea Vehicle (MAUV) project [22] and was adapted for the NASA/NBS Standard Reference Model Telerobot Control System Architecture (NASREM) being used on the space station Flight Telerobotic Servicer [23]. The basic building block of RCS-3 is shown in Figure 2(c). The block diagram of NASREM is shown in Figure 3. The principle new features introduced in RCS-3 are the World Model and the operator interface. The inclusion of the World Model provides the basis for planning. Each of the TD modules in a RCS-3 hierarchy have a planner and an executor for each of the subsystems under its control. The planners generate state-tables (or their equivalent) for the executors. The planners and executors provide temporal task decomposition. Each TD module also contains a job assignment submodule to provide spatial task decomposition.



(a)
Basic Building Block of RCS-1

(c)
Basic Building Block of RCS-3



(b)
Basic Building Block of RCS-2

(d)
Basic Building Block of RCS-4

Figure 2. The basic building blocks of the RCS control paradigm.

NASREM: NASA/NBS STANDARD REFERENCE MODEL

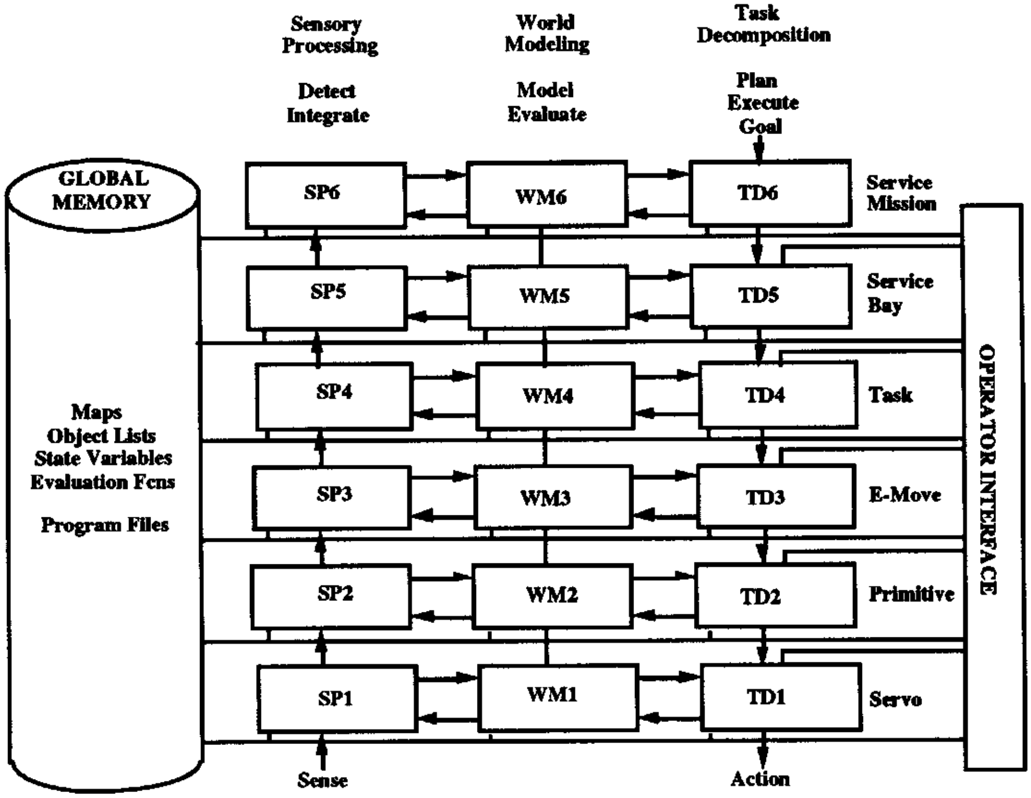


Figure 3. The NASREM (RCS-3) Hierarchy. It is composed of three types of computing modules: Task Decomposition (TD), World Modeling (WM), and Sensory Processing (SP). These are interconnected to each other through a communications network, and serviced by a Global Memory knowledge base. There is an Operator Interface to each level.

RCS-4 is under current development by the NIST Robot Systems Division. The basic building block is shown in Figure 2(d). The principle new feature in RCS-4 is the Value Judgment (VJ) system. VJ modules provide to the RCS-4 control system the type of functions provided to the biological brain by the limbic system. The VJ modules evaluate perceived situations, attach values to perceived entities and outcomes, and indicate what is important. The VJ modules also evaluate predicted results of hypothesized plans. The RCS-4 design also includes a much more fully developed set of interactions between the SP and WM modules than any of its precursors.

RCS-4 is designed to address complex applications where high bandwidth communications are impossible, such as autonomous vehicles operating on the battlefield, deep undersea, or on distant planets. These applications require autonomous value judgments and sophisticated real-time perceptual capabilities. RCS-3 will continue to be used for less demanding applications such as manufacturing, construction, or telerobotics for near-space or shallow undersea operations where communication bandwidth is less restricted.

A Machining Workstation Example

Figure 4 illustrates how the RCS-3 system architecture can be applied to a specific machining workstation consisting of a machine tool, a robot, and a part buffer. RCS-3 produces a layered graph of processing nodes, each of which contains a Task Decomposition (TD), World Modeling (WM), and Sensory Processing (SP) module. These modules are richly interconnected to each other by a communications system. The global memory knowledge base and operator interface are not shown in Figure 4.

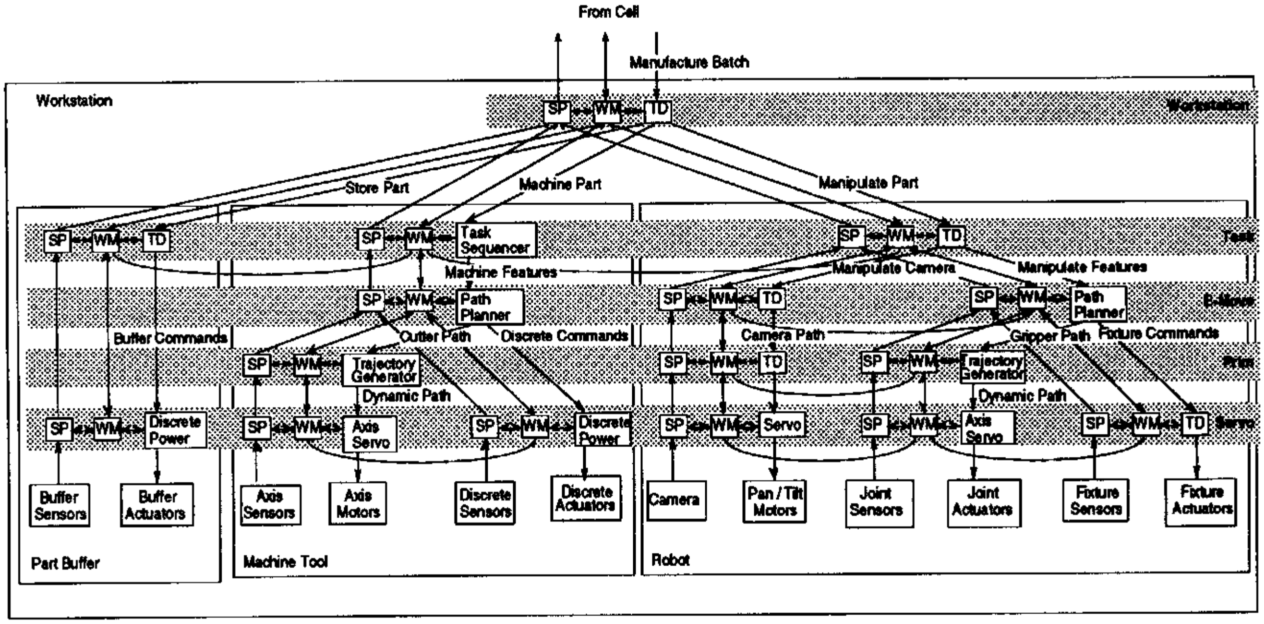


Figure 4. A RCS-3 application of a machining workstation containing a machine tool, part buffer, and robot with vision system.

The Task Decomposition (TD) modules support planning, job assignment, and plan execution. Each TD module decomposes commands from a higher level into a sequence of subcommands, or a plan, for one or more subsystems at the next lower level. At each level, plans have shorter spatial and temporal horizons. Tasks such as production scheduling, with long planning horizons, occur at the higher levels, while tasks such as dynamic motion paths, with short horizons, occur at the lower levels. In cases where a single lower-level TD module can receive commands from more than one supervisor, a priority scheme is used to resolve conflicts. In cases where several TD modules require simultaneous use of a single resource, conflicts may be resolved by arbitration.

Task decomposition consists of both spatial and temporal decomposition. Spatial decomposition partitions a task into jobs to be performed by different subsystems. Spatial task decomposition results in a tree structure, where each node corresponds to a TD module, and each branch of the tree corresponds to a communication link in the chain of command. Temporal decomposition partitions each job assigned to a subtree into sequential subtasks along the time line. The result is a set of subtasks, all of which when accomplished, achieve the task goal.

The Sensory Processing (SP) modules are responsible for gathering data from sensors, filtering and integrating this data, and interpreting it. Noise rejection techniques such as Kalman filtering are implemented here, as well as feature extraction, pattern recognition, and image understanding. At the upper levels of the hierarchy, more abstract interpretations of sensory data are performed and inputs from a variety of sensor systems are integrated over space and time into a single interpretation of the the external world.

The global memory Knowledge Databases (KD) (not shown in Figure 4) forms a passive, hierarchically structured, data store. The World Modeling (WM) modules provide the operations that act upon the KD, and can be considered the active portion. The KD may contain a variety of information, such as maps, object lists, state variables, evaluation functions, and program files containing plans or scripts, stored across diverse platforms. The WM has the capability to perform abstract operations on the KD, such as drawing inferences, making predictions, and evaluating situations. At higher levels, WM maps cover larger areas with lower resolution; points, lines, and surfaces are clustered into objects; objects are clustered into groups; and state variables, inferences, and predictions become more abstract.

The communications pathways and computing modules define a layered graph, or lattice, of nodes and directed arcs. Each layer contains one or more nodes which correspond to computational subdivisions of the sensory-motor system. Each node contains a TD, WM, and SP module.

Arcs connecting the nodes correspond to communication paths. Horizontal arcs carry information such as questions, answers, and shared data. Downward flowing vertical arcs carry commands, priorities, and database configuration information. Upward flowing vertical arcs carry status and processed sensory information. Output from the bottom level TD modules drive actuators. Input to the bottom level SP modules convey data from sensors.

Figure 4 illustrates the ability of RCS-3 to integrate discrete sensors such as microswitches with more complex sensors such as cameras and resolvers. Discrete commands can be issued to valves and fixtures, while continuous signals are provided to servoed actuators. Notice that in some branches of the control tree, levels may be absent. For example, in the case of the part buffer, discrete commands at the Task level can be directly executed by the Servo level. In the case of the part fixture, discrete commands issued from the robot E-Move level can be executed by the Servo level.

The branching of the control tree (for example, between the camera and manipulator subsystems of the robot), may depend on the particular algorithm chosen for decomposing a particular task. The specifications for branching reside in the task frame (defined later) of the current task being executed. Similarly, the specification for sharing information between WM modules at a level also are task dependent. In Figure 4, the horizontal curved lines represent the sharing of state information between subtrees in order to synchronize related tasks. The information that must be shared is also dependent on the specific choice of task decomposition algorithms defined in the task frame.

Figure 5 summarizes the relationship in RCS-4 between the organizational hierarchy that is defined by the command tree, the computational hierarchy that is defined along each chain of command, and the behavioral hierarchy that is produced in state-space as a function of time. The behavioral hierarchy consists of state/time trajectories that are produced by computational modules executing tasks in real-time.

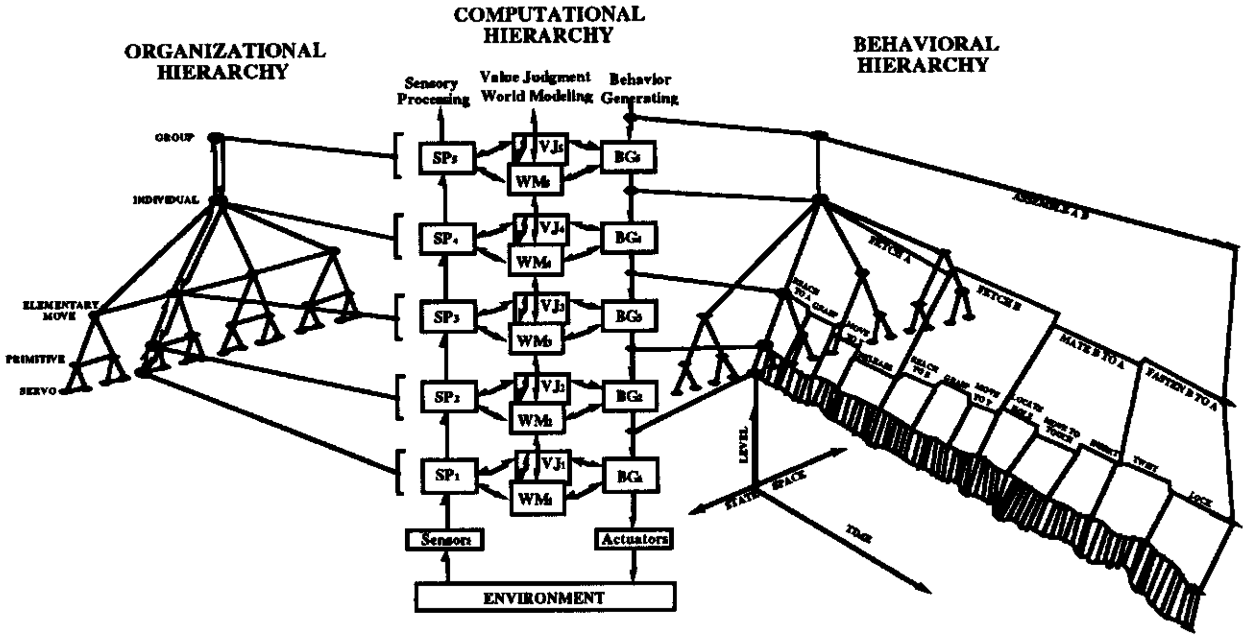


Figure 5. The relationship between the organizational hierarchy of subsystems, the computational hierarchy of computing modules, and behavioral hierarchy of state trajectories that result as the computing modules execute tasks. The behavioral hierarchy illustrates the decomposition of an assembly task <ASSEMBLE A,B>.

Timing

Levels in the RCS command hierarchy are defined by temporal and spatial decomposition of goals and tasks into levels of resolution, as well as by spatial and temporal integration of sensory data into levels of aggregation. Temporal resolution is manifested in terms of loop bandwidth, sampling rate, and state-change intervals. Temporal span is measured in length of historical traces and planning horizons. Spatial resolution is manifested in the resolution of maps and grouping of elements in subsystems. Spatial span is measured in range of maps and the span of control.

Figure 6 is a timing diagram that illustrates the temporal relationships in a RCS hierarchy containing seven levels of task decomposition and sensory processing. The sampling rate, the command update rate, the rate of subtask completion, and the rate of subgoal events, increases at the lower levels of the hierarchy, and decreases at upper levels of the hierarchy. The particular numbers shown in Figure 6 simply illustrate the relative timing between levels. Specific timing requirements are dependent on specific machines and specific applications.

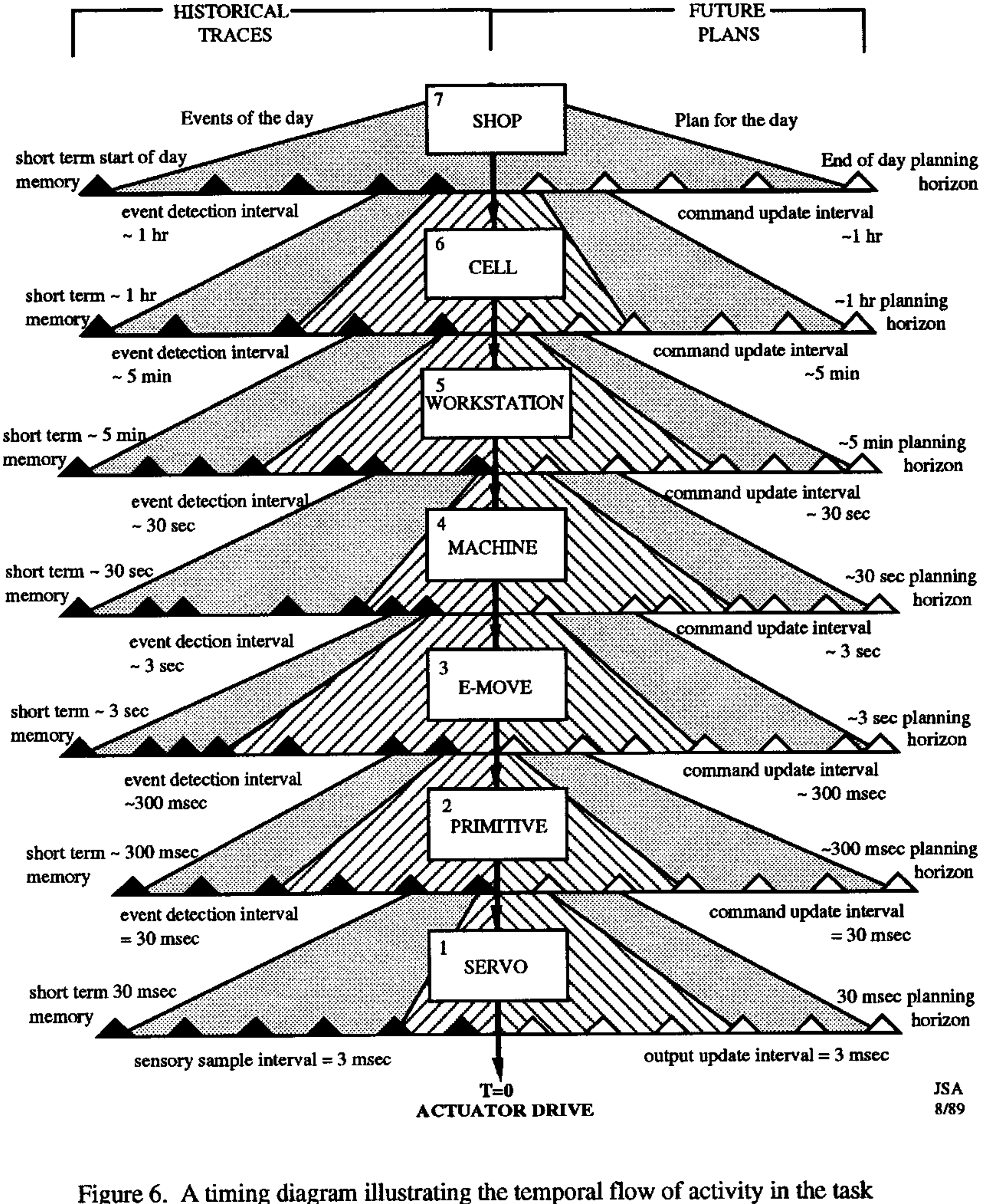


Figure 6. A timing diagram illustrating the temporal flow of activity in the task decomposition and sensory processing systems.

A fundamental principle of RCS-3 and 4 is that planning goes on simultaneously and continuously at all levels. At each level, planners periodically generate plans containing, on average, five to ten steps. At each successive level, the first one or two steps in the current plan from the level above are further decomposed into subplans with an order of magnitude more resolution, and an order of magnitude less span in time and space. As a result, the average period between task commands decreases by about an order of magnitude at each lower level.

Replanning is done either at cyclic intervals, or whenever emergency conditions arise. The cyclic replanning interval may be as short as the average command update interval at each level, or about about ten percent of the planning horizon at each level. This allows the real-time planner to generate a new plan about as often as the executor generates a new output command. Less frequent replanning will reduce the computational speed requirements for real-time planning, but may also reduce control system effectiveness.

There exists a duality between the task decomposition and the sensory processing hierarchies, as can be seen in Figure 6. A sensory event at one hierarchical level can be defined as a sequence of events at the next lower level. At each level of the hierarchy, the sensory processing modules look back into the past about as far the planner modules look forward into the future. At each level, future plans have about the same detail as historical traces.

Functionality

The functionality of each level can be derived from the characteristic timing of that level. For example in a manufacturing environment:

Level 7 -- Shop

The shop level schedules and controls the activities of one or more cells for an extended period (for example 24 hours or more). At the shop level, orders are sorted into batches and commands are issued to the cell level to develop a production schedule for each batch. At the shop level, the world model symbolic database contains names and attributes of orders and the inventory of tools and materials required to fill them. Maps describe the location of, and routing between, manufacturing cells.

Level 6—Cell

The cell level schedules and controls the activities of several workstations for a period about ten times shorter than the cell planning horizon (for example, one or two hours). Batches of parts and tools are scheduled into workstations, and commands are issued to workstations to perform machining, inspection, or material handling operations on batches or trays of parts. The world model symbolic database contains names and attributes of batches of parts and the tools and materials necessary to manufacture them. Maps describe the location of, and routing between, workstations.

Level 5—Workstation

The workstation level schedules tasks and controls the activities within each workstation with a still shorter planning horizon (for example about a five minutes). A workstation may consist of a group of machines, such as one or more closely coupled machine tools, robots, inspection machines, materials transport devices, and part and tool buffers. Plans are developed and commands are issued to equipment to operate on material, tools, and fixtures in order to produce parts. The world model symbolic database contains names and attributes of parts, tools, and buffer trays in the workstation. Maps describe the location of parts, tools, and buffer trays.

Level 4—Equipment task

The equipment level schedules tasks and controls the activities of each machine within a workstation with (for example) about a 30 second planning horizon. (Tasks that take much longer may be broken into about 30 second segments at the workstation level.) Level 4 decomposes each equipment task into elemental moves for the subsystems that make up each piece of equipment. Plans are developed that sequence elemental movements of tools and grippers, and commands are issued to move tools and grippers so as to approach, grasp, move, insert, cut, drill, mill, or measure parts. The world model symbolic database contains names and attributes of parts, such as their size and shape (dimensions and tolerances) and material characteristics (mass, color, hardness, etc.). Maps consist of drawings that illustrate part shape and the relative positions of part features.

Level 3—Elemental move (E-move)

The E-move level schedules and controls simple machine motions requiring (for example) a few seconds, such GO-ALONG-PATH, MOVE-TO-POINT, MILL-FACE, DRILL-HOLE, MEASURE-SURFACE, etc. (Motions that require more time may be broken up at the task level into several elemental moves.) Plans are developed and commands are issued that define safe path waypoints for tools, manipulators, and inspection probes so as to avoid collisions and singularities, and assure part quality and process safety. The world model symbolic database contains names and attributes of part features such as surfaces, holes, pockets, grooves, threads, chamfers, burrs, etc. Maps consist of drawings that illustrate feature shape and the relative positions of feature boundaries.

Level 2—Primitive

The primitive level plans paths for tools, manipulators, and inspection probes so as to minimize time and optimize performance. It computes tool or gripper acceleration and deceleration profiles taking into consideration dynamical interaction between mass, stiffness, force, and time. Planning horizons are (for example) on the order of 300 milliseconds. The world model symbolic database contains names and attributes of linear features such as lines, trajectory segments, and vertices. Maps (when they exist) consist of perspective projections of linear features such as edges, lines, and approach–grip trajectories.

Level 1—Servo level

The servo level transforms commands from tool path to joint actuator coordinates, interpolates between primitive trajectory points with (for example) a 30 millisecond look ahead. It servos individual actuators and motors to the interpolated trajectories. Position, velocity, or force servoing may be implemented, and in various combinations. Commands that define actuator torque or power are output (for example) every 3 milliseconds. The servo level also controls the output drive signals to discrete actuators such as relays and solenoids. The world model symbolic database contains values of state variables such as joint positions, velocities, and forces, proximity sensor readings, position of discrete switches, condition of touch probes, as well as image attributes associated with camera pixels. Maps consist of camera images and displays of sensor readings.

At the Servo and Primitive levels, the command output rate is perfectly regular. At the E-Move level and above, the command output rates typically are irregular because they depend on the geometry of the world, and are event driven.

Summary/Conclusion

The RCS paradigm has been developed over a number of years, has evolved through several versions, and has been applied to many different applications. These include laboratory robotics, the NIST Automated Manufacturing Research Facility, a Field Material Handling Robot, Multiple Autonomous Undersea Vehicles, TMAP, TEAM, and Robotic Testbed military land vehicle projects, the space station Flight Telerobotic Servicer, Standard Reference Model Architecture for Coal Mine Automation [24]. RCS has served as a model for the Next Generation Controller, as a guideline for Submarine Operational Automation Systems, and as implementation software for Post Office Automation. Many researchers and engineers have found the RCS paradigm useful as a reference model architecture for intelligent machine systems. A large number of papers have been published on RCS, giving details of the structure and function of the TD (or BG), WM, SP, KD, and VJ modules, and describing how different versions of RCS have been implemented for a variety of applications [25-35].

Current NIST research is directed towards two goals: 1) To formalize the RCS paradigm so that it can serve as the basis for interface standards in open-systems architectures for real-time intelligent control systems, and 2) To develop a methodology for implementing intelligent control systems for a wide variety of applications.

REFERENCES

[1] L.D. Erman, F. Hayes-Roth, V.R.Lesser, and D.R. Reddy, D.R., "Hearsay-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty", *Computer Survey,* vol. 23, June 1980, pp. 213-253.

[2] J.E. Laird, A. Newell, and P. Rosenbloom, "SOAR: An Architecture for General Intelligence", *Artificial Intelligence*, vol. 33, 1987, pp. 1-64.

[3] Honeywell Inc., "Intelligent Task Automation Interim Technical Report II-4", December 1987.

[4] J. Lowerie, et al. "Autonomous Land Vehicle", *Annual Report, ETL-0413*, Martin

Marietta Denver Aerospace, July 1986.

[5] D. Smith and M. Broadwell, "Plan Coordination in Support of Expert Systems Integration", *Knowledge-Based Planning Workshop Proceedings*, Austin, TX, December 1987.

[6] J.R. Greenwood, G. Stachnick, H.S. Kaye, "A Procedural Reasoning System for Army Maneuver Planning", *Knowledge-Based Planning Workshop Proceedings*, Austin, TX, December 1987.

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

[8] R. Brooks, "A Robust Layered Control System for a Mobile Robot", *IEEE Journal of Robotics and Automation*, vol. RA-2, 1, March 1986.

[9] G.N. Saridis, "Foundations of the Theory of Intelligent Controls", *IEEE Workshop on Intelligent Control,* 1985.

[10] A. Meystel, "Intelligent Control in Robotics", *Journal of Robotic Systems*, 1988.

[11] Barbera, A.J., J.S. Albus, M.L. Fitzgerald, "Hierarchical Control of Robots Using Microcomputers," Proceedings of the 9th International Symposium on Industrial Robots, Washington, DC, March 1979.

[12] J.S. Albus, "Mechanisms of Planning and Problem Solving in the Brain", *Math. Biosciences*, 45, p.247-293, 1979.

[13] 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, 1983.

[14] J.S. Albus, C. McLean, A.J. Barbera, and M.L. Fitzgerald, "An Architecture for Real-Time Sensory-Interactive Control of Robots in a Manufacturing Environment", *4th IFAC/IFIP Symposium on Information Control Problems in Manufacturing Technology*, Gaithersburg, MD, October 1982.

[15] C. McLean, M. Mitchell, E. Barkmeyer, "A Computer Architecture for Small Batch Manufacturing," IEEE Spectrum, May 1983, p. 59.

[16] M. Mitchell, E. Barkmeyer, "Data Distribution in the NBS Automated Manufacturing

Research Facility", Proc. of the IPAD2 Conf., Denver, April 1984.

[17] C. McLean, "Process Planning Research at the AMRF," Navy Manufacturing Technology Report, November 1984, p.5.

[18] C. McLean, "An Architecture for Intelligent Manufacturing Control," 1985 ASME Computers in Engineering Conf., Boston, August 1985.

53] J. Albus, C. McLean, A. Barbera, M. Fitzgerald, " An Architecture for Real-Time Sensory-Interactive Control of Robots in a Manufacturing Environment," 4th IFAC/IFIP Symposium on Information Control Problems in Manufacturing Technology, Gaithersburg, October 1982.

[19] C. McLean, H. Bloom, T. Hopp, "The Virtual Manufacturing Cell," IFAC/IFIP Conf. on Information Control Problems in Manufacturing Technology, Gaithersburg, MD., October 1982.

[20] McCain, H.G., R.D. Kilmer, S. Szabo, A. Abrishamian, "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.

[21] Szabo, S., H.A. Scott, K.N. Murphy, S.A. Legowik, "Control System Architecture for a Remotely Operated Unmanned Land Vehicles," Proceedings of the Fifth IEEE International Symposium on Intelligent Control, Philadelphia, PA, September 5-7, 1990.

[22] Albus, J.S., "System Description and Design Architecture for Multiple Autonomous Undersea Vehicles", National Institute of Standards and Technology, Technical Report 1251, Gaithersburg, MD, September 1988.

[23] Albus, J.S., H.G. McCain, and R. Lumia, "NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)," National Institute of Standards and Technology, Technical Report 1235, Gaithersburg, MD , April 1989.

[24] Huang, H.M., R.Quintero, "Task Decomposition Methodology for the Design of a Coal Mining Automation Hierarchical Real-Time Control System," Proceedings of the Fifth IEEE International Symposium on Intelligent Control, Philadelphia, PA, September 5-7, 1990.

[25] Albus, J.S., “RCS: A Paradigm for Intelligent Control”, IEEE Intelligent Controls

Conference, 1992.

[26] Albus, J.S., “Outline for a Theory of Intelligence”, IEEE SMC (in publication).

[27] Fiala, J.C., "Manipulator Servo Level Task Decomposition," National Institute of Standards and Technology, Technical Note 1255, October 1988.

[28] Wavering, A.J., "Manipulator Primitive Level Task Decomposition," National Institute of Standards and Technology, Technical Note 1256, September 1988.

[29] Kelmar, L., "Manipulator Servo Level World Modeling," National Institute of Standards and Technology, Technical Note 1258, December 1989.

[30] Kelmar, L., "Manipulator Primitive Level World Modeling," National Institute of Standards and Technology, Technical Note 1273, December 1989.

[31] Kelmar, L, R. Lumia, "World Modeling for Sensory Interactive Trajectory Generation," Proceedings of the Third International Symposium on Robotics and Manufacturing, Vancouver, BC, Canada, July 18-20, 1990.

[32] Fiala, J.C., R. Lumia, "An Approach to Telerobot Computing Architecture," National Institute of Standards and Technology, Internal Report NISTIR 90-4357, June 1990.

[33] Michaloski, J., T. Wheatley, "System Factors in Real-Time Hierarchical Control," National Institute of Standards and Technology, Internal Report NISTIR 90-4386, August 1990.

[34] Fiala, J.C., "Note on NASREM Implementation," National Institute of Standards and Technology, Internal Report NISTIR 89-4215, March 1990.

[35] Fiala, J.C., "Interfaces to Teleoperation Devices," National Institute of Standards and Technology, Technical Note 1254, October 1988.