NATO Advanced Workshop on
Robots and Biological Systems

A CONTROL ARCHITECTURE FOR COOPERATIVE INTELLIGENT ROBOTS

by James S. Albus
Robot Systems Division
National Institute of Standards and Technology
Gaithersburg, MD
USA

Abstract

Intelligent behavior requires a control system architecture that ties together actuators, sensors, sensory processing, task decomposition, world modeling, goal selection, and value judgments into an integrated system. A hierarchically layered architecture with horizontal communications within layers is proposed.

Intelligent cooperative behavior requires common goals, common strategies, agree upon division of labor, and a common view of the world. To the extent that all individuals in a population are working on the same top level input command, they share a common goal. To the degree that all have similar task vocabularies and control programs resident in their task decomposition modules, there exists a basis for common strategy. To the extent that all agree on the structure of the command tree, there exists a basis for division of labor. To the degree that individuals have the same information resident in their world model, they share a common view of the world. When all these exist simultaneously, there exists the basis for intelligent cooperative behavior. Communication is a mechanism for maintaining a common goal, strategy, division of labor, and world view.

1. Introduction

During the past decade, the problem of cooperative intelligent agents has become a substantial subfield of artificial intelligence. Distributed artificial intelligence, sometimes known as distributed problem solving [1] is concerned with distributed planning and control as well as with distributed data and multi-processor computing architectures. The techniques of distributed problem solving have been most extensively studied in the areas of speech understanding [2], manufacturing process scheduling [3,4] air traffic control [5,6], and multiple autonomous undersea vehicles [7].

Nilsson [8] has suggested that distributed problem solving may be crucial to understanding of artificial intelligence. There are many reasons to believe that it is crucial to understanding of biological intelligence as well. The brain is not a single computer nor is mind a single process [9]. The brain is a community of billions of neurons, and the mind a collection of millions of processes all interconnected and independently executing simultaneously. The brain and nervous system is one vast distributed processing system with interconnected modules for task decomposition, world modeling, sensory processing, and value estimating.

Furthermore, problem solving in the brain is not limited to the individual. The nature of the world is such that few creatures can survive alone, and there are many survival advantages in cooperation between individuals. There is safety in numbers, and many tasks, from hunting to escaping predation, are more likely to succeed if pursued by cooperating groups rather than individuals.

Virtually all animals exhibit some form of cooperative behavior. Fish swim in schools, birds congregate in flocks, grazing animals form herds, and many predators hunt in packs. The advantages are that the school or flock always has many eyes looking in many directions at once for danger. The herd can form a circle to ward off predators. The hunting pack has many members searching for prey and cooperating in its capture.

In higher animals, cooperative behavior is extensive, even pervasive. Parents cooperate in feeding offspring, protecting them from danger, and training them in the ways of the world. Among the most intelligent of creatures such as elephants, apes, and humans, families live together for years, and group interactions involve elaborate protocols including the establishment of rank in social dominance hierarchies.

In human societies, more efficiently organized social structures with extensive division of labor and well trained workers tend to be more efficient than less organized cultures, surpassing them economically and dominating them militarily and politically.

It is anticipated that artificially intelligent autonomous military vehicles will be most effective when deployed in groups.

2. Elements of Cooperation

Cooperation between intelligent individuals requires a set of commonalties. There must be a common goal, a common strategy or plan for achieving that goal, a common view or model of the world, an agreed upon division of labor, and a means for communication between the cooperating individuals.

Common goals

Among creatures of nature, all goals relate ultimately to survival and gene propagation. Survival requires detecting and escaping predators, and hunting for food. Propagation requires acquiring and defending territory, attracting a mate, defeating rivals, and feeding and defending offspring.

Where there exists common goals, such as to watch for and ward off predators, to track and kill prey, or to defend offspring, cooperation will be the predominant behavior. Where the goal of one individual conflicts with that of another, such as in the acquisition of territory, or attracting breeding partners, conflict and competition will predominate.

Common strategy

Without a common strategies, common goals cannot be achieved. A group cannot stay together unless all members move in roughly the same direction. A nesting pair will not be successful in raising a family unless they have a common strategy where one protects the nest while the other searches for food. A hunting pack will fail unless the members maintain contact and cooperate in the kill.

Division of labor

Cooperation implies that each member plays a role in a joint activity. Group tasks are decomposed into subtasks to be carried out by individual group members. In caring for young, one parent may hunt for food, while the other protects the nest. In defending young, one parent may attract attention to itself by attacking the intruder or feigning injury, while the other member sits motionless on the nest to provide camouflage.

In complex human enterprises such as manufacturing, construction, or military operations, division of labor is carried to a high degree with many hierarchical levels of decomposition and many different specialized activities.

Common world model

All intelligent creatures carry some internal representation of the external world, either explicitly or implicitly. Without a common world model representation of critical information such as the distribution of danger, the direction of escape, and the relative positions of friendly agents, cooperation cannot take place.

Communication

Communication is one means by which common goals, strategies, division of labor, and representation of the world can be established. An individual may use communication to announce its own identity, position, and state of mind, including intent. Communication can be used to sound an alarm at the presence of danger, or to announce the detection of prey. Communication can also be used to command or request action from others, or simply to inform others of events or state changes in the world.

Communication implies that information is: 1) encoded, 2) transmitted, 3) received, and 4) understood. Without all four, communication has not taken place. However, communication does not require intent on the part of the sender to send a message to the receiver. In fact, communication is often quite unintentional, and preventing an enemy from intercepting communication between friendly agents can be a major concern.

Language is the means by which information is encoded. Language has three basic components: vocabulary, syntax, and semantics. Vocabulary is the set of words in the language. Words may be represented by symbols. Syntax, or grammar, is the set of rules for stringing together symbols to form sentences. Semantics is the set of rules for combining words into meaningful patterns, or messages. Messages are sentences that convey useful information.

Communication and language are by no means unique to human beings. Virtually all creatures, even insects, communicate in some way, and hence have some form of language. For example, many insects transmit messages announcing their identity and position. Sometimes this is done acoustically, sometimes by smell. The goal may be to attract a mate, or to permit recognition and/or location by other members of a group. Of course, insects have very little other information to communicate, and hence their language has only one or two of what might be called words, with little or no grammar, and very simple semantics.

The encoding and transmission of information can take many forms, including body posture, gestures, facial expressions, hair erection, and acoustic signals generated by variety of mechanisms from stamping the feet or clapping the hands, to snorts, squeals, chirps, cries, and shouts. Depending on its complexity, a language may be capable of communicating many messages, or only a few. More intelligent creatures have a larger vocabulary and more complex grammar. They are also better able to understand and act on the meaning of messages.

Communication of danger signals improves the survival probability of all individuals in the group. However, communication is only advantageous to those individuals who are able to recognize the danger messages. Those benefit most who are the quickest and most discriminating in the recognition of danger messages, and most effective in responding with appropriate action.

In general, the benefit, or value, of communication is roughly proportional to the product of the amount of information contained in the message, multiplied by the ability of the receiver to understand and act on that information, multiplied by the importance of the act to survival and gene propagation. Greater intelligence enhances both the individual's and the group's ability to analyze the environment, to encode and transmit information about it, to detect messages, to recognize their significance, and act effectively on information received. Greater intelligence produces more complex languages capable of expressing more information, i.e. more messages with more shades of meaning.

In social species, communication also provides the basis for societal organization and division of labor. Communication of threats that warn of aggression can help to establish the social dominance hierarchy, and reduce the incidence of physical harm from fights over food, territory, and sexual partners. Communication of alarm signals indicate the presence of danger, and in some cases, identify its type and location. Communication of pleas for help enables group members to come to each other's assistance or defense. Communication between members of a hunting pack enable them to hunt more effectively by cooperating as a team in the the tracking and killing of prey.

Among humans, the most basic form of communication is facial expressions, gestures, body language, cries, tone of voice, and pantomime. However, the human brain is capable of generating ideas of much greater complexity and subtlety than can be expressed through a few cries and gestures. In order to transmit messages commensurate with the complexity of human thought, human languages have evolved grammatical rules capable of stringing words from vocabularies consisting of thousands of entries into sentences which express ideas and concepts with exquisitely subtle nuances of meaning. During this evolutionary process, the human vocal apparatus has developed complex mechanisms for making a variety of sounds that can be strung together to generate an infinite number of messages in a spoken language.

In any species, language evolves to support the level of communication generated by the intelligence of that species. Cooperation and language are thus products of intelligence, and if we wish to fully understand either, we must first understand the mechanisms of intelligence itself.

Of course, our understanding of intelligence is far from complete and will remain so for a long time, but much is known that is relevant to the issue of cooperative behavior. For example, the fundamental elements of intelligence, and the basic functional relationships between them, are shown in Figure 1.

3. The Elements of Intelligence

a. ACTUATORS -- Within any intelligent system there are actuators which move, exert forces, and position arms, legs, hands, and eyes. Actuators generate forces to point sensors, excite transducers, move manipulators, handle tools, and steer 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 -- Sensors may include visual brightness and color, tactile, force, torque, position, 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

AN INTELLIGENT MACHINE SYSTEM

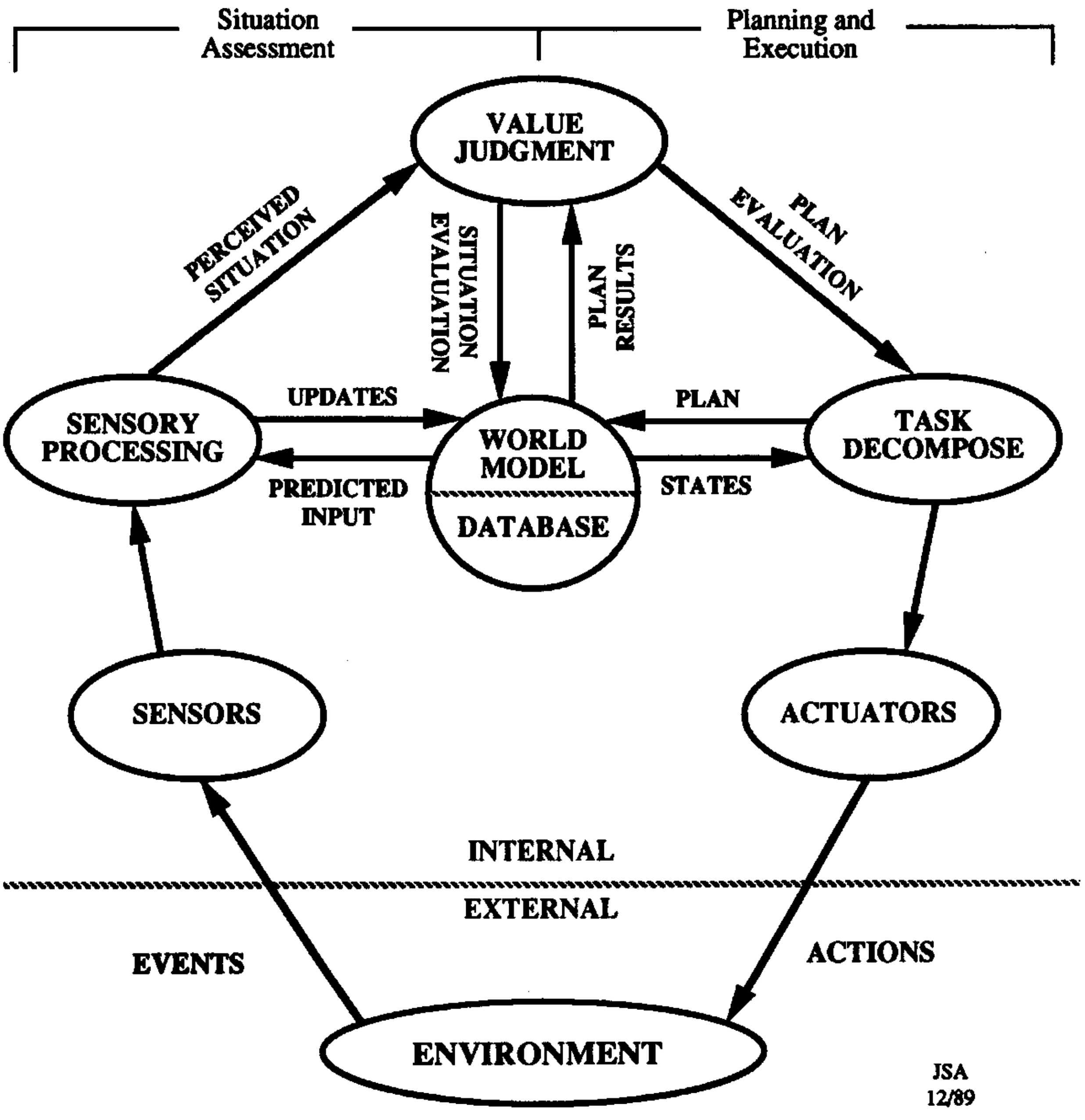


Figure 1. The elements of intelligence and the functional relationships between them.

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 are fused into a consistent unified perception of the state of the world. Sensory processing algorithms compute distance, shape, orientation, surface characteristics, and material properties of objects and regions of space. Sensory processing may include recognition of speech and interpretation of language and music.

d. TASK DECOMPOSITION -- Behavior is generated in a task decomposition system that plans and executes tasks by decomposing them into subtasks, and sequencing these subtasks so as to achieve goals. Goals are selected 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.

e. 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.

f. 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 pursuing one object and fleeing from another. Without a value system, any biological creature would soon be eaten by others, or destroyed by its own inappropriate actions. The value system also computes the probability of correctness and assigns believability and uncertainty parameters to state variables.

4. An Architecture for Intelligent Machine Systems

Each of the above elements of intelligent systems are reasonably well understood. However, intelligence is 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 an intimate and sophisticated way. A number of system architectures have been proposed, and a few have been implemented. [10, 11, 12, 13].

The particular architecture that will be discussed here has been partially implemented in a number of versions of the Real-time Control System (RCS) developed over the past 13 years at the National Institute for Standards and Technology (NIST formerly NBS). The first version of RCS was developed by Barbera for laboratory robotics and adapted by for the NIST Automated Manufacturing Research Facility (AMRF) [13]. RCS-2 was implemented and installed in the Horizontal machining workstation of the AMRF [14-18], and in the Cleaning and Deburring workstation [19]. RCS was adapted to the AMRF Vertical machining workstation [20, 21] and to the Inspection workstation [22]. The AMRF communications and database systems were developed [23], as well as the RCS emulator [24]. RCS-2 was adapted to the Army Field Material-handling Robot (FMR) [25] .

RCS-3 has been, and is being, used on a number of additional projects, including the NBS/DARPA Multiple Autonomous Undersea Vehicle (MAUV) project and the Army TEAM

(Technology Enhancement for Autonomous Vehicles) semi-autonomous land vehicle project. RCS-3 also forms the basis of the NASA/NBS Standard Reference Model Telerobot Control System Architecture (NASREM) [26] being used on the space station Flight Telerobotic Servicer.

The version of RCS described in this paper organizes the elements of intelligence defined in Section 3 into a six layer hierarchy of computing modules, partitioned into four vertically integrated sections: task decomposition (TD), world modeling (WM), sensory processing (SP), and value judgment (VJ). All the modules in this hierarchy are richly interconnected to each other by a fifth vertical section, the global memory (GM) which provides intermodule communication, as illustrated in Figure 2. In a biological brain, the function of the GM modules are provided by neurons as shown in Figure 3. In artificial systems, the physical implementation of global memory may be a common memory, a blackboard, or a message passing system, or some combination thereof.

Hierarchical vs. Heterarchical

Figure 4 elaborates the architecture of Figure 2 so as to show both the hierarchical and heterarchical (horizontal) relationships involved.

The architecture is organized as a hierarchical tree in the sense that commands and status feedback flow hierarchically up and down the chain of command. It is also hierarchical in that sensory information is processed into increasingly higher levels of abstraction, and that information stored at various levels of resolution in the world model is organized hierarchically.

The levels in this hierarchical tree structure are defined by the decomposition of tasks into levels of spatial and temporal resolution. Spatial resolution is manifested in the span of control and in the resolution and range of maps. Temporal resolution is manifested in terms of loop bandwidth, sampling or state-change intervals. Temporal span is measured in length of historical traces, and future planning horizons.

High level commands, or goals, are decomposed both spatially and temporally through a hierarchy of control levels into strings and patterns of subcommands. Each task decomposition TD module represents a node in a command tree. Each command node receives input commands from one and only one supervisor, and outputs a temporal string of subcommands to one or more subordinate modules at the next level down in the tree. Outputs from the bottom level consist of drive signals to motors and actuators. Figure 5 shows a single branch of the hierarchical tree structure of Figure 4, seen from a high level looking down to a single motor unit.

The proposed architecture is, however, also heterarchical (or horizontal) in the sense that data is shared horizonally between heterogeneous modules at the same level. At each hierarchical level, the architecture is horizontally interconnected by wide bandwidth communication pathways between task decomposition TD, world modeling WM, sensory processing SP, and value judgment VJ modules.

The JA, PL, and EX submodules in the task decomposition TD module communicate voluminously with each other and with the world modeling module at the same level. They negotiate with each other and resolve constraints. They ask What If? questions of the WM for planning and What Is? questions for execution.

WM modules are constantly in communication with SP modules at their corresponding level, predicting sensory input, and being updated by the observed state of the world. WM modules also respond to "What is?" and "What if?" questions from the executors and planners in the TD modules at their corresponding level.

Thus, although the proposed architecture incorporates a command and control hierarchy in the form of a formal logical tree, there exists an extensive horizontal flow of non-command

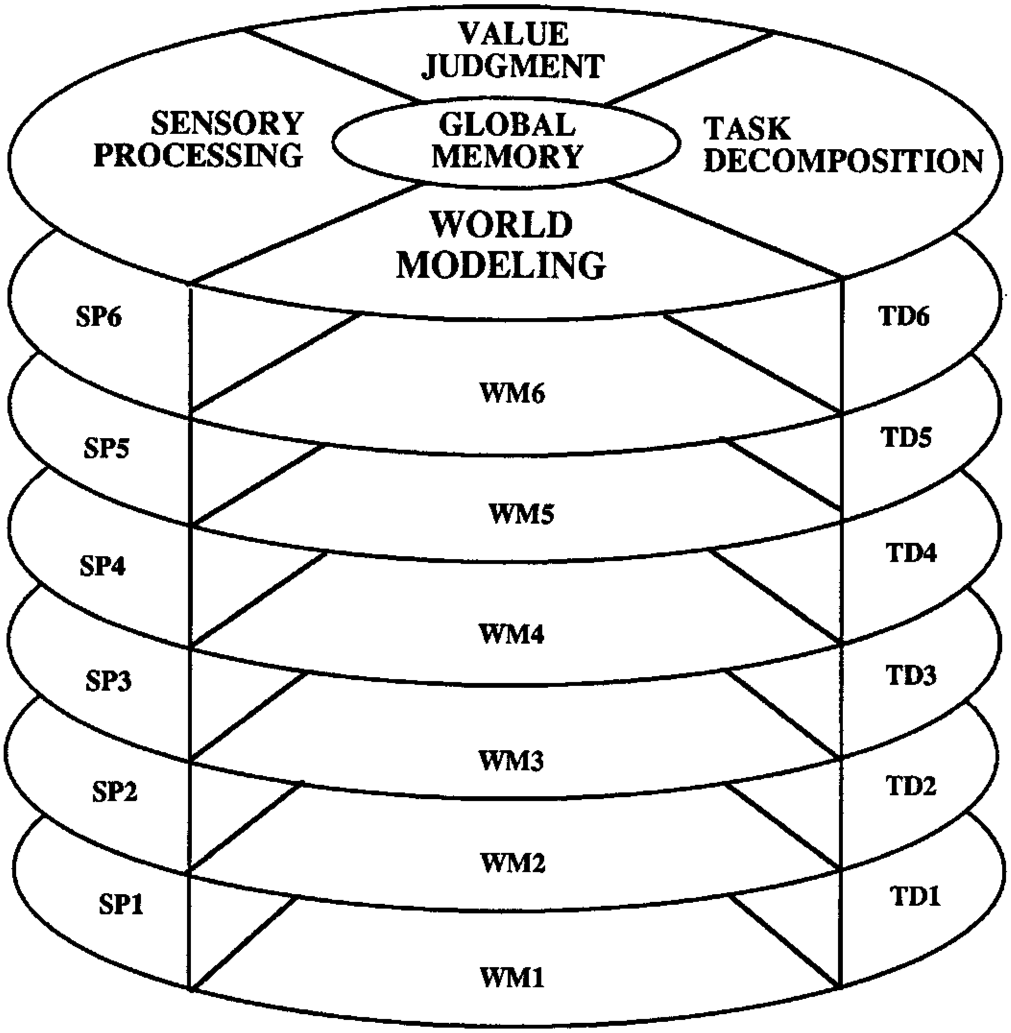


Figure 2. A six layer version of RCS partitioned into four vertically integrated sections: task decomposition TD, world modeling WM, sensory processing SP, and value judgment VJ. These are interconnected through the global memory GM which acts as a communications system.

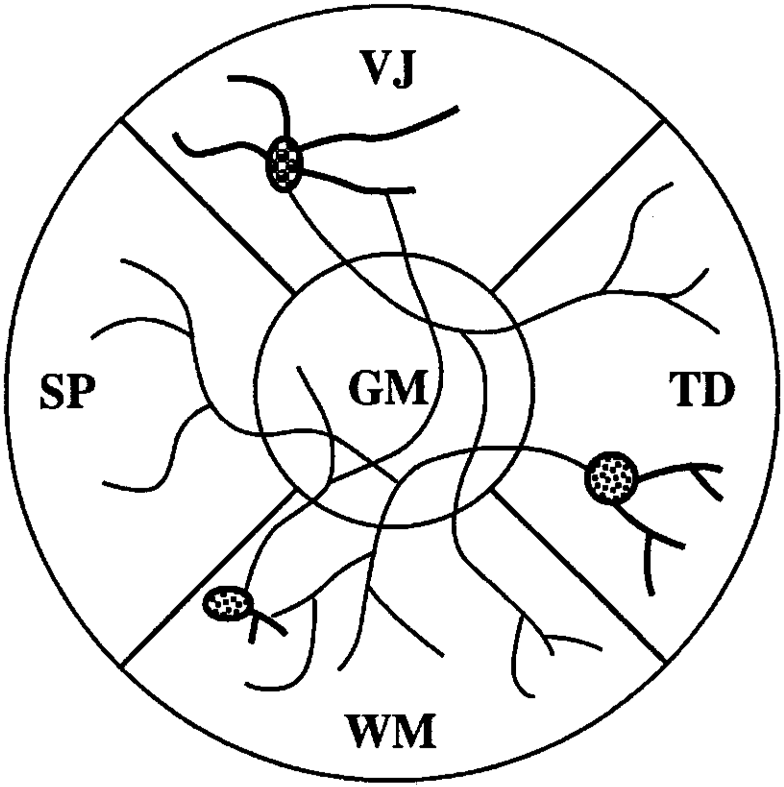


Figure 3. Neuronal pathways from a neuron in the TD module to one in the WM module to one in the VJ module and back to the TD module. Axon collaterals also go from the TD neuron to the SP module, and from the VJ neuron to the WM module. All pathways go through the GM module.

modules at their corresponding level.

Thus, although the proposed architecture incorporates a command and control hierarchy in the form of a formal logical tree, there exists an extensive horizontal flow of non-command information between TD, WM, SP, and VJ modules at the same level. In fact, the volume of information flowing horizonally completely dwarfs the amount flowing vertically.

The horizontal flow of information is, however, largely confined within the same subtree. The requirement for communications is much less between modules in separate subtrees. This is indicated by the number of horizontal lines in Figure 4.

It should be emphasized that global memory is not a single physical memory. In RCS, global memory is distributed over a number of single board computers and memory cards, and in many cases, these are distributed among separate vehicles.

Task Decomposition

For any intelligent system, there exists a set of tasks which the system knows how to do. Each task in this set can be assigned a name. The task vocabulary is the set of tasknames assigned to the set of tasks the system is capable of performing.

Task knowledge is knowledge of how to perform a task; plus information as what tools, materials, time, resources, and conditions are required; and what are the expected costs, benefits, and risks. A task frame is a data structure in which task knowledge is stored. For each taskname in the task vocabulary, there exists a task frame, or data structure, of the form:

TASKNAME	-- name of the task
a) actor	-- agent performing the task
b) action	-- activity to be performed
c) object	-- thing to be acted upon
d) goal	-- event that successfully terminates the task
e) requirements	-- tools, time, resources, and materials needed -- conditions that must be satisfied to begin -- information that may be required
f) procedures	-- a state graph defining the plan (or schema) -- functions that may be called -- algorithms that may be needed
g) effects	-- expected results of task execution -- expected costs, risks, benefits -- estimated time to complete

Task frames are essential to task planning. They are used by the task planners for generating hypothesized actions. They are used by the world model in predicting the results of hypothesized actions.

Task knowledge is typically difficult to discover, but once known, can be readily transferred to others. Task knowledge may be acquired by trial and error learning, but more often it is acquired from a teacher, or from written or programmed instructions. In most cases, the ability to successfully plan complex tasks is more dependent on the amount of task knowledge stored in the task frame than on the sophistication of a planner in reasoning about tasks.

For example, the common household task of preparing a food dish is typically performed by following a recipe. A recipe is an informal task frame for cooking. Gourmet dishes rarely result from reasoning about possible combinations of ingredients, still less from trial and error combinations of food stuffs. Recipes often are closely guarded secrets that, once published, can easily be understood and followed by others.

ORGANIZATIONAL HIERARCHY

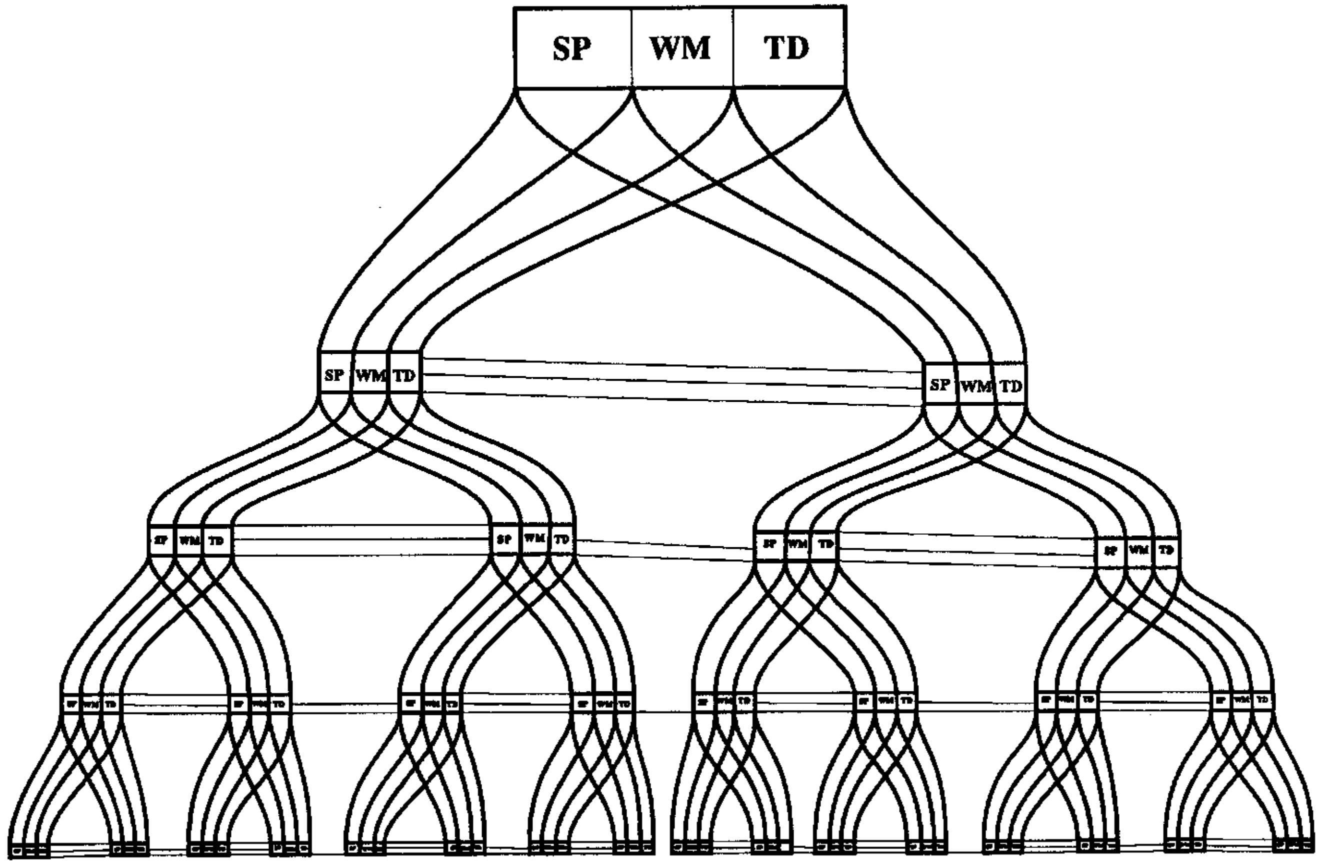


Figure 4. The organizational structure of RCS showing the spatial decomposition into subsystems and the horizontal communications between computing modules at the same level.

COMPUTATIONAL HIERARCHY

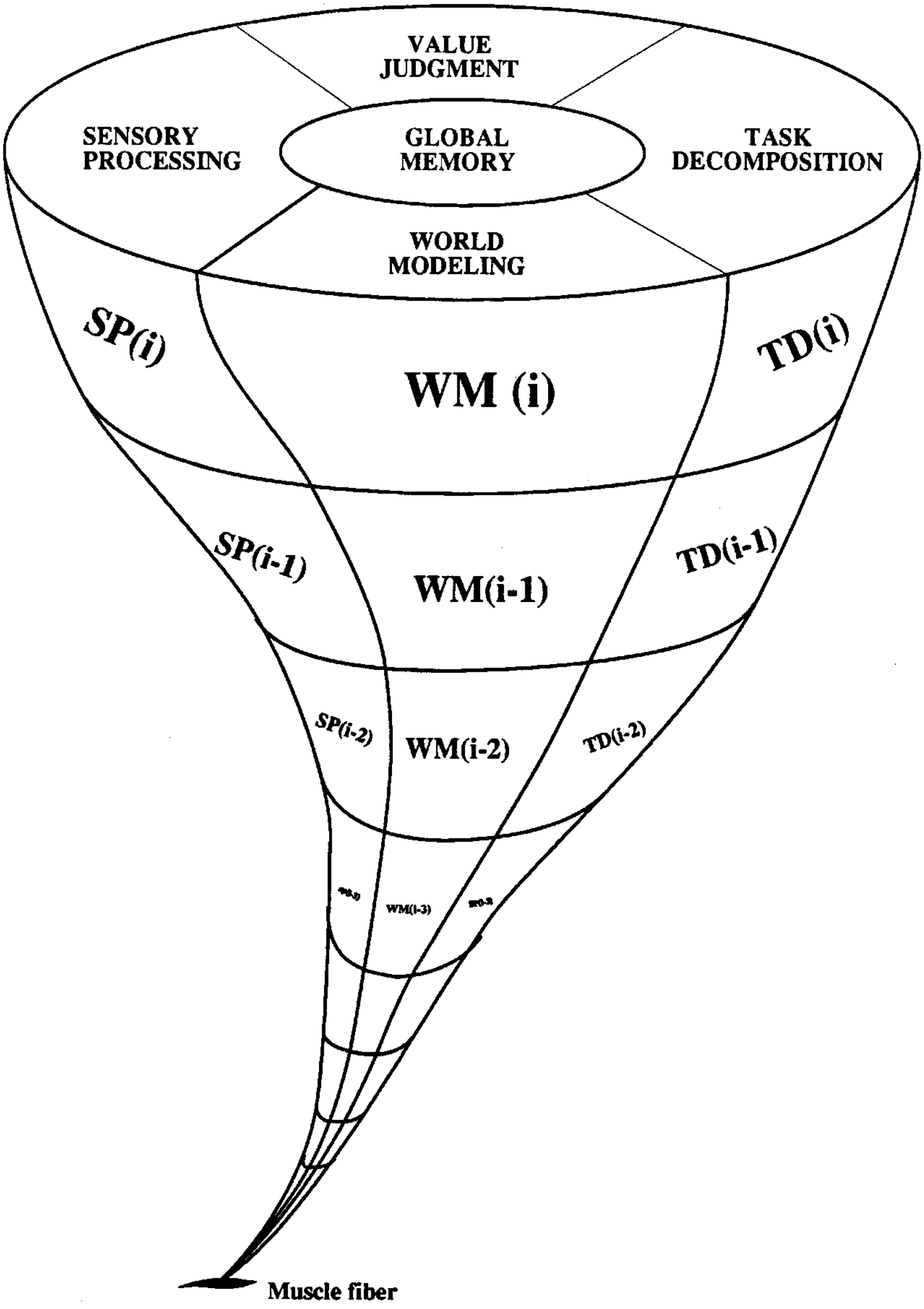


Figure 5. A single branch of the hierarchical tree of Figure 4, seen from a high level looking down to a single motor unit.

Task Decomposition (TD) Modules

TD modules in the task decomposition hierarchy plan and execute the decomposition of high level goals into low level actions. Task decomposition involves both a spatial decomposition (into concurrent actions by different subsystems), and a temporal decomposition (into sequential actions along the time line).

Each TD module at each level consists of three sublevels as shown in Figure 6. These three sublevels function to decompose the input task into both spatially and temporally distinct subtasks.

1) the job assignment sublevel -- JA submodule

The JA submodule is responsible for spatial task decomposition. It partitions the input task command into N spatially distinct jobs to be performed by N physically distinct subsystems, where N is the number of subsystems controlled by the TD module.

The JA submodule is also responsible for assigning tools and allocating physical resources (such as arms, hands, legs, sensors, tools, and materials) to each of its subordinate subsystems for their use in performing their assigned jobs. These assignments are not necessarily static. For example, the job assignment submodule at the individual level may assign the arms to the manipulation subsystem in response to a <use tool> task command, and later assign the arms to the locomotion subsystem in response to a <swim> task command.

The job assignment submodule also selects the coordinate system in which the task is to be performed.

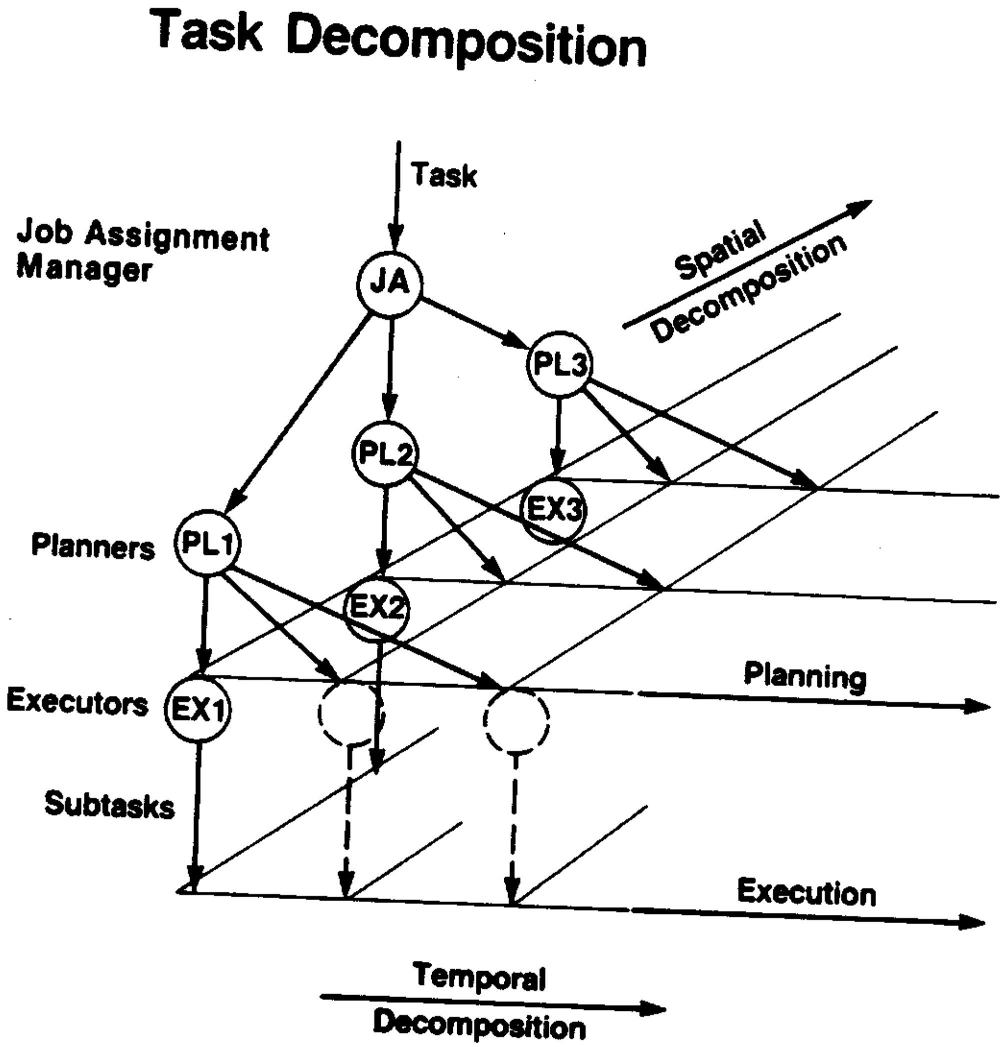
2) the planner sublevel -- planners PL(j) $j = 1, 2, \ldots, N$

For each of the N subsystems, there exists a planner submodule PL(j). Each planner submodule is responsible for decomposing the job assigned to its subsystem into a temporal sequence of planned subtasks. Task planning typically requires evaluation of alternative hypothetical sequences of subtasks. Each planner PL(j) functions by hypothesizing some action or series of actions. The WM module then predicts the effects of the action(s), and the VJ module computes the value of the resulting expected states of the world, i.e. a VJ evaluation function performs a priority-weighted cost-benefit analysis on the results predicted by the WM module. The evaluation values returned to the PL(j) planner are sorted to select the hypothetical sequence of actions producing the best evaluation. This becomes the plan to be executed by EX(j). This sequence corresponds to the communication flow depicted in Figure 3.

A job command to a subsystem may contain constraints on time, or specify job-start and job-goal events. A job assigned to one subsystem may also require synchronization with other jobs assigned to different subsystems. These constraints and requirements are specified by, or derived from, the task frame. Each planner PL(j) submodule is responsible for checking its plan against plans generated by each of the other N-1 planners at the same level to determine if there are mutually conflicting constraints. If a conflict is found, constraint relaxation algorithms [27] may be applied until a solution is found. If no solution can be found, the planner reports failure to the job assignment submodule, and a new job assignment may be tried.

It must be noted that planning does not necessarily require reasoning from first principles as often presumed by AI planners. Planning can be accomplished by selecting from a library of partially or completely prefabricated plans [28], scripts [29], or schema [30].

For example, nature has provided many biological creatures with an extensive library of genetically determined plans, sometimes known as instinct. The range of behavior that can be



Task Decomposition

Figure 6. The job assignment JA module performs a spatial decomposition of the task. The planners PL(j) perform temporal decomposition. Each executor EX(j) closes a real-time control loop that servos the subtask to the plan.

generated by a library of instinctive plans at each hierarchical level, with each plan containing a number of conditional branches and error recovery routines, can be extremely large and complex. For many species, this library of plans is adequate to assure the survival of the species.

3) an executor sublevel -- executor submodules EX(j).

There is an executor EX(j) for each planner PL(j). The executor submodules are responsible for successfully executing the plans prepared by their respective planners. When an executor is informed by the world model that a subtask in its current plan is successfully completed, the executor steps to the next subtask in that plan. When all the subtasks in the current plan are successfully executed, the executor steps to the first subtask in the next plan. If the feedback indicates the failure of a planned subtask, the executor branches immediately to a preplanned emergency subtask. Its planner simultaneously begins work selecting or generating an error recovery sequence which can be substituted for the former plan which failed. Output subcommands produced by executors at level i become input commands to job assignment submodules in TD modules at level i-1.

Planners PL(j) constantly operate in the future, each generating a plan to the end of its planning horizon. The executors EX(j) always operate in the present, at time t=0, constantly monitoring the current state of the world reported by feedback F from the world model. In a sampled data control system, the executor modules operate on short, regular intervals, or execution cycles. The length of the execution cycle is set by a system state clock. The period of the state clock is defined by the rate at which sensory input data is sampled. At each state-clock cycle, each executor submodule compares the current step in its plan against the current state of the world, and computes an output subcommand designed to null the difference. The executors at each level react to feedback in one state clock period.

Thus, at each level, each executor submodule closes a reflex arc, or servo loop, and the executor submodules at the various hierarchical levels form a set of nested servo loops. In order to prevent instability due to nested servo loop interaction, the executor loop bandwidth decreases about an order of magnitude at each higher level.

Sensory Processing - SP modules

Sensory processing SP modules filter, correlate, and integrate sensory information over both space and time so as to detect, recognize, and measure patterns, features, objects, events, and relationships in the external world. This is shown in Figure 7. The SP modules consist of four sublevels:

1) A set of comparator modules
that compare sensor observations with world model predictions

2) A set of temporal integrators
that integrate similarities and differences over time

3) A set of spatial integrators
that integrate similarities and differences over space

4) A threshold module
that recognizes objects and detects events

The SP modules are dual to the TD modules. Whereas the TD modules decompose tasks spatially and temporally into subtasks for multiple subsystems, SP modules integrate data from multiple sources over extended time intervals into unified patterns, or gestalt perceptions of the world.

Similarities, differences, and newly detected or recognized events, objects, and relationships are

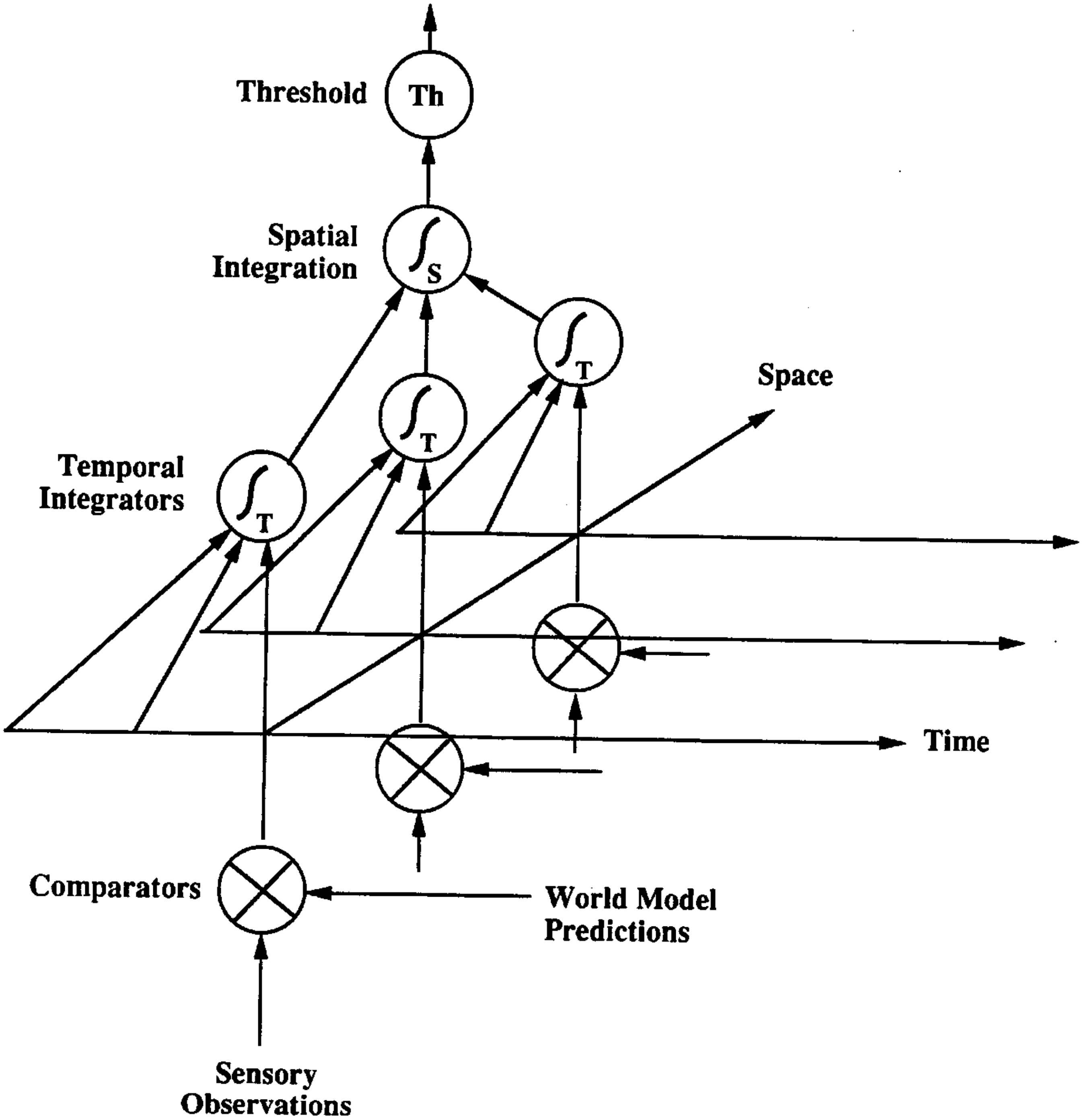


Figure 7. Each sensory processing SP module consists of: 1) a set of comparators that compare sensory observations with world model predictions, 2) a set of temporal integrators that integrate similarities and differences, 3) a spatial integration that fuses information from different sensory data streams, and 4) a threshold detector that recognizes entities and detects events.

entered by the WM modules into the world model global memory database. Objects or relationships perceived to no longer exist are removed.

World Modeling - WM modules

The world model is the system's internal best estimate of the history, current state, and possible future states of the world, including the states of the system being controlled. The world model includes both the WM modules and a knowledge base stored in global memory.

As shown in Figure 8, the WM modules at each level perform the following functions:

a) Answer "What if?" questions asked by the job assignment and planner submodules in the corresponding level TD modules.

b) Answer "What is?" questions asked by the JA, PL, and EX submodules in the corresponding level TD modules. The job assignment and planner submodules use this information for planning. The executors use it to monitor and servo the task.

c) Provide predictions of expected sensory input to the corresponding SP modules, based on the state of the task and estimates of the external world.

d) Maintain the global memory knowledge base, keeping it current. The WM modules update the knowledge base based on correlations and differences between WM predictions and SP observations.

Contents of Global Memory

The knowledge in the global memory consists of maps, lists, and state variables.

a) Maps are 2-dimensional data structures that show the relative position of objects and regions. Maps may also contain overlays, that may indicate values such as utility, cost, risk, etc. assigned to regions or objects on the map. These values are computed by the VJ modules, and can be used both for planning and executing tasks.

Maps may be stored in two types of coordinate frames: world coordinates, and egosphere coordinates. These are illustrated in Figure 9.

A world coordinate map is a two dimensional representation in which latitude and longitude are typically the x-y coordinates, and each pixel contains a pointer to a data structure that gives the physical properties and z-dimension of the region or objects covered by that pixel. A vehicle moving through the world can be represented as an object moving on the world map. The world map may be scrolled so as to keep a particular vehicle of interest at the center.

An egosphere is a polar coordinate system map centered on the sensor system. Pixels are referenced by azimuth and elevation, and each pixel contains a pointer to a data structure that gives the physical properties and range of the region or object covered by that pixel. As the system moves through the world, the contents of the pixels on the egosphere change. Thus, the egosphere must be periodically recomputed to be current.

Each hierarchical level may use a different resolution working copy of a portion of the world map. The working copy at each level covers a region that bounds the task being planned at that level with resolution such that subtasks being planned at that level are easily resolved. Each level may also use a different set of egosphere overlays.

b) Entity lists

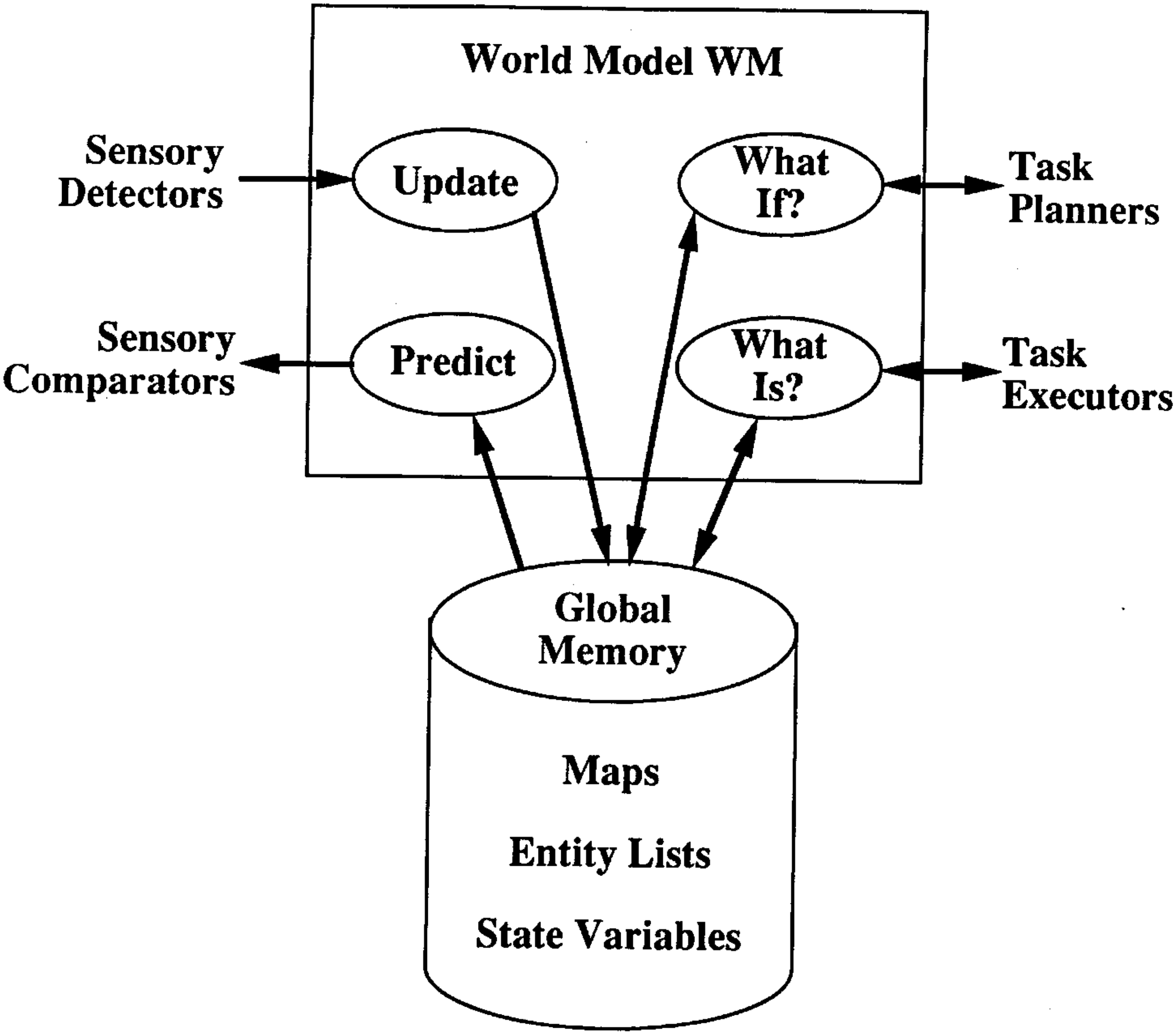
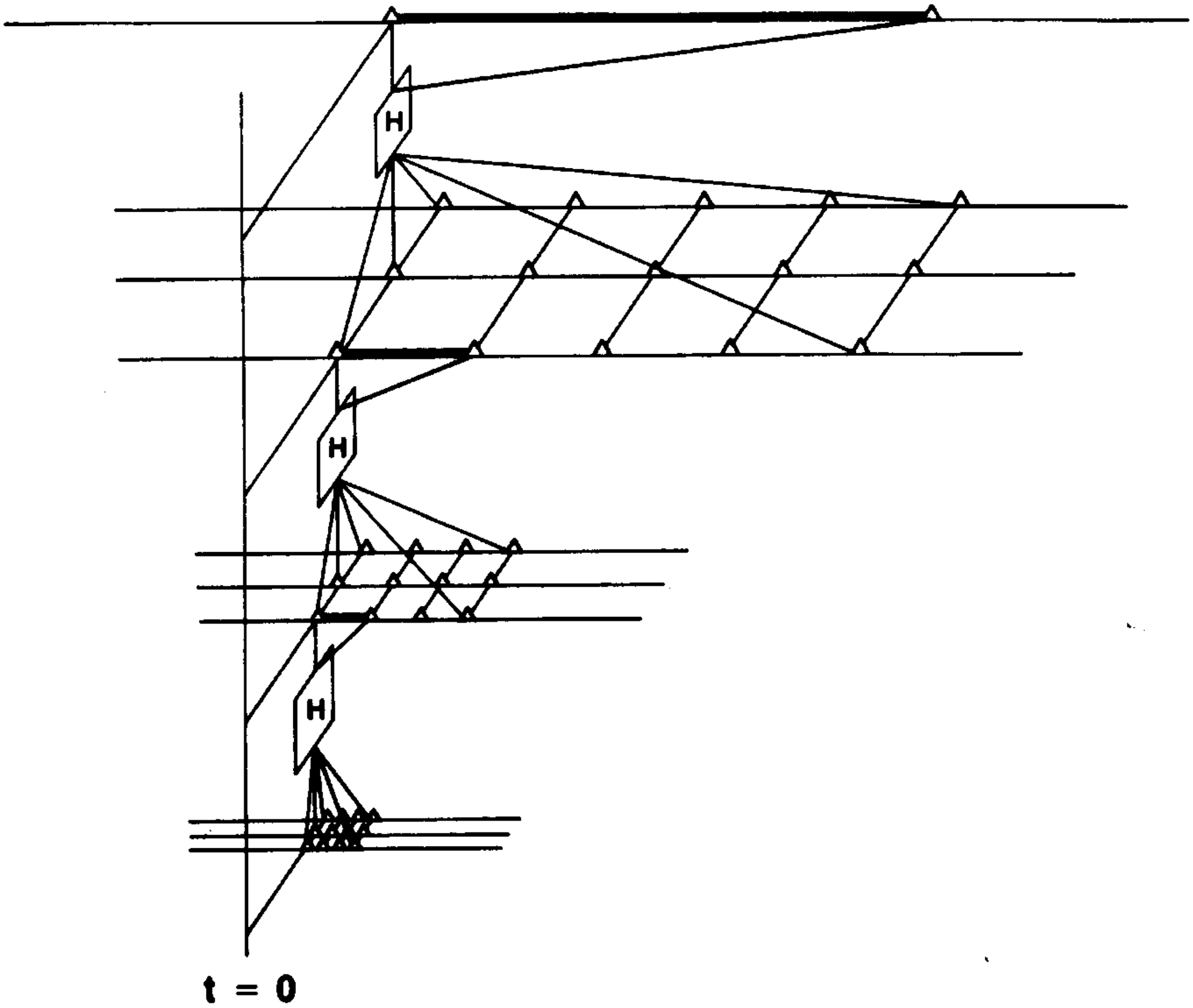


Figure 8. At each level, the world model responds to What If? and What Is? queries from task planners and executors, updates the global memory, and predicts sensory input.



Hierarchical Planning

Figure 9. Geometric relationships between world map and egosphere coordinates.

All entities of interest, such as objects, tasks, features, regions, and relationships, and events, are listed in the global memory database, indexed by name and characteristic features. Each entity in the list has a data form, or "frame", containing attributes. For example an object frame might have the form:

ENTITY NAME	-- Name
Type	-- (Object, event, task, region, etc.)
	-- (Specific or generic)
Geometry	-- Shape
	-- Size
	-- Sub-entity list
	-- Parent entity
Position	-- World map coordinates
	-- Egosphere coordinates
Dynamics	-- Velocity
Linear	
Rotational	-- Acceleration
Attributes	-- Model or class
	-- Mass
	-- Color
	-- Substance
	-- Typical behavior (of intelligent entities)
	-- Speed
	-- Range
Value state variables	-- (Attract-repulse, etc.)
	-- Confidence levels

For moving objects, entity frames may contain not only current map coordinates, but a past history or trace of coordinate positions. Event frames contain information such as start and end time, duration, type, cost, payoff, etc. At different hierarchical levels, object frames have different levels of detail and spatial resolution, and event frames have different levels of temporal resolution.

c) State Variables

State variables may be entries in entity frames, or may define values for maps overlays. The entire set of state variables define the system's best estimate of the state of the world, including both the external environment and the internal state of the TD, WM, SP, and VJ modules.

Value Judgments -- VJ modules

The value judgment VJ modules contain functions that enable them to evaluate events, situations, or objects observed by sensory input, or stored in the world model, or predicted from the world model. VJ modules also contain functions which compute confidence factors and probabilities of recognized events, and statistical estimates of stochastic state variables.

Inputs to the VJ modules are state variables, goal priorities, and values assigned to entities such as vehicles, targets, and resources.

Outputs from the VJ modules are value state-variables, such as good, bad, cost, risk, payoff, attractive, repulsive, etc., that may be assigned to entity frames or map regions. For example, attraction or repulsion values may be assigned to objects, places, or persons. Values of cost, risk, and payoff may be assigned to events such as subtask completion, information acquisition, and vehicle survival. Cost and risk values may also be associated with map route segments.

VJ modules also compute subgoal priorities, and assign value state-variables such as cost and benefit to subtask goals for evaluating plans and choosing execution options. For example, in planning the planners hypothesize actions, and the WM modules generate predictions of what will result from hypothesized actions. The VJ modules apply evaluation functions to these results and return value state-variables to the planners. These values can then be assigned to nodes in a game tree as the planners conduct a search over the space of possible futures. The planners then choose the sequence of hypothesized actions that produce the best evaluation as the plan to be executed.

Executors may also use value state-variables from the VJ modules to select among various possible next actions in a plan state- graph. Value state-variables computed by the VJ modules on the current state of the world can be assigned to variables in the plan graph edge predicates, to be used by the executors to make moment by moment behavioral decisions.

In biological creatures, the VJ modules reside in the limbic system.

Timing

A timing diagram illustrating the temporal flow of activity in the task decomposition and sensory processing systems is shown in Figure 10. Temporal decomposition along the time axis produces a "dechunking" of tasks into subtasks according to their characteristic time intervals for execution. These characteristic time intervals provide the primary criteria for defining hierarchical levels in the control architecture.

For example,

At level 1, feedback inputs and command outputs occur on the order of every few milliseconds. Command inputs occur around every 30 milliseconds. Smooth motion is planned for each actuator, and servo loops control position, velocity, and force.

At level 2, feedback inputs and command outputs occur about every 30 milliseconds. Command inputs occur around every 300 milliseconds. Dynamically coordinated motion is planned and controlled for tightly coupled groups of actuators.

At level 3, feedback inputs and command outputs occur about every 300 milliseconds. New command inputs occur around every 3 seconds. Obstacles are avoided and clear path trajectories are planned and executed for entire subsystems such as manipulation and locomotion.

At level 4, feedback inputs and command outputs occur about every 3 seconds. Command inputs change on average about every 30 seconds. Simple tasks on single entities are planned and sequenced for the whole individual.

At level 5, feedback inputs and command outputs occur about every 30 seconds. Command inputs change on average about every 5 minutes. Complex tasks by one or more actors on multiple entities are planned and sequenced for the self group. In each individual, level 5 defines that individual's role in the group.

At level 6, feedback inputs and command outputs occur about every 5 minutes. Command inputs change on average about every hour. Tasks by one or more groups on other groups are planned and executed. In each individual, level 6 defines that individual's role in intergroup activities.

In the temporal decomposition hierarchy, each higher level produces about an order of magnitude increase in planning horizon and decrease in plan resolution along the time line. The rate of subtask execution specifies the loop bandwidth requirements for the feedback control loop through each level. When combined with vehicle velocity, it also defines the extent of the spatial regions over which tasks are planned and executed. Thus, temporal decomposition produces not only a functional task decomposition, but a hierarchical layering of control loops and a hierarchical

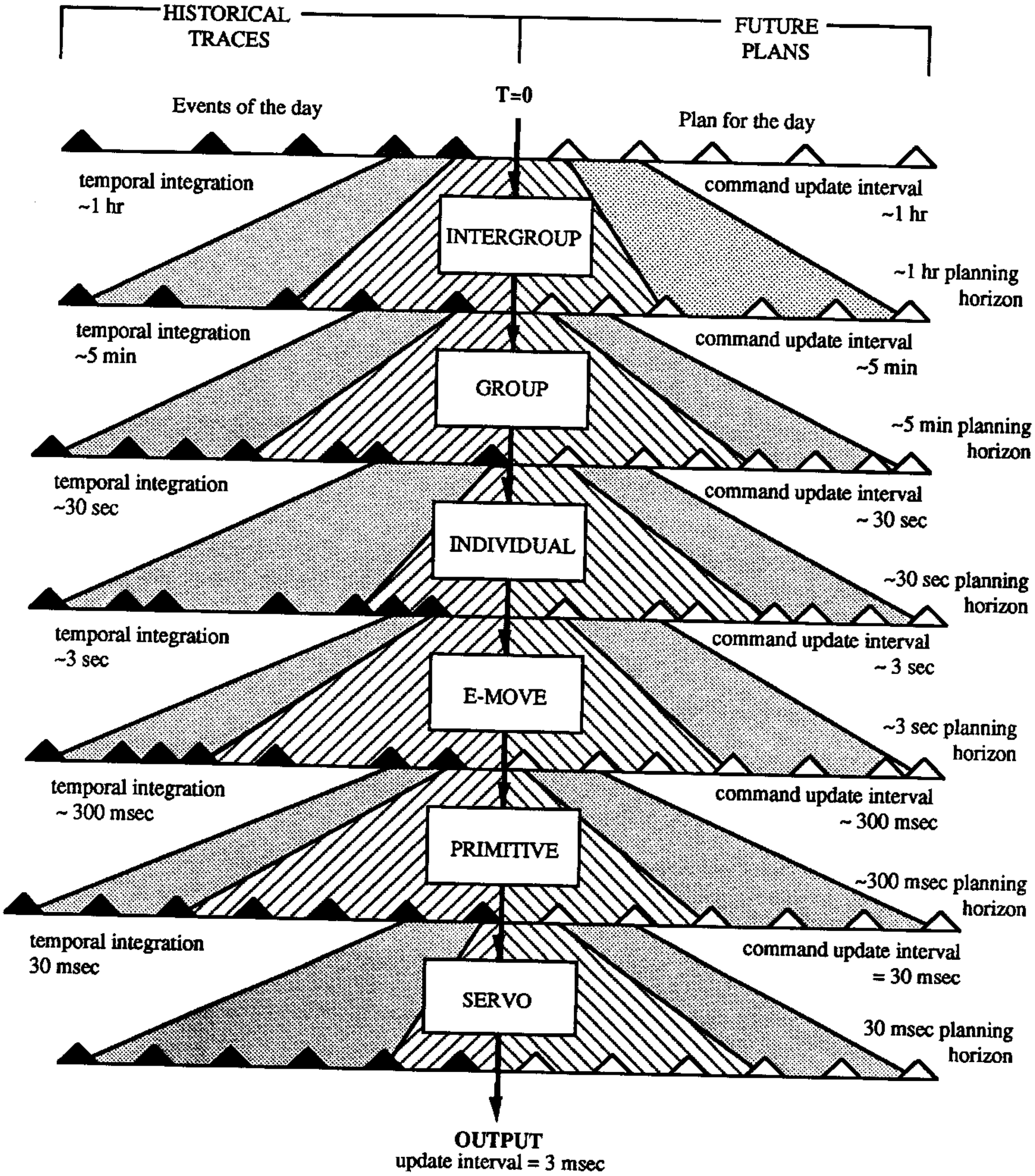


Figure 10. A timing diagram illustrating the temporal flow of activity in the task decomposition and sensory processing systems. Input to the intergroup level is derived from a top level executor where high level sensory events and circadian rhythms react with habits and daily routines that constitute the plan for the day.

structuring of world model maps and entities that support the temporal and functional decompositions.

In the model presented here, input to the intergroup level is derived from a top level executor which operates on a top level plan. The top level plan may be determined by instinct and habits of daily routine, or it may be developed by a top level planner that generates or selects plans based on VJ evaluations. Command input to the top level planner consists of the single unchanging goal, <propagate genes>.

Feedback input to the top level executor consists of high level events and state variables such as hormone levels, circadian rhythms, drives of hunger, thirst, and sex, the recognition of objects and relationships between objects, and the detection of danger, or other emotionally stimulating events.

At each lower level, plans are formulated (or selected) in real- time to accomplish the current and next task in the plan of the level immediately above. Each task in the higher level plan is decomposed into a lower level plan of at least two, and on average, about 10 subtasks. The planning horizon thus shrinks exponentially by about an order of magnitude at each successively lower level of the hierarchy.

Figure 11 shows a simple example of three levels of planning activity represented in Gantt notation. The input command to the top level TD module is decomposed by the job assignment manager and three planners of the top TD module into three simultaneous plans consisting of four subtasks each. The first executor of the top level TD module outputs the current subtask command in its plan to a second level TD module. This second level task command is decomposed by the job assignment manager and three planners in the second level TD module into three plans, again consisting of four subtasks each. The first of the second level executors outputs the current activity in its plan to a third level TD module, which further decomposes it into three plans of four subtasks. At each level the final subgoal events in the plans correspond to the goal of the input task. At each successively lower level, the planning horizon becomes shorter, and the subtasks become more detailed and fine structured.

The timing ratio between levels is not at regular, particularly at the higher levels, because executor outputs are event driven, triggered by feedback from the environment. Depending on events in the world, the rate of change of subtasks may vary considerably. Furthermore, an output from any level may be repeated any number of times. Thus, the apparent interval between outputs may vary over an even wider range.

Figure 10 also illustrates the duality between the task decomposition and the sensory processing hierarchies. At each level in the hierarchy, the sensory processing modules look back into the past about as far as the planner modules look forward into the future. At each level, future plans have about the same detail as historical traces.

5. Cooperation Between Multiple Individuals

In a factory control system, where high bandwidth communication is always available between all control modules, there can be a single group control module for each group, and a single intergroup control module to coordinate the activities of the various groups.

In the case of autonomous individuals, however, communication may not always be available, or may be corrupted by noise, or be limited in bandwidth. In order to cope with these situations, each autonomous individual should contain its own group and intergroup control levels. This enables individuals to execute cooperative group and intergroup tasks with a minimum of communication.

If the level 5 and 6 TD modules of each individual contain the same algorithms and task

WORLD MAP/EGOSPHERE TRANSFORMATION

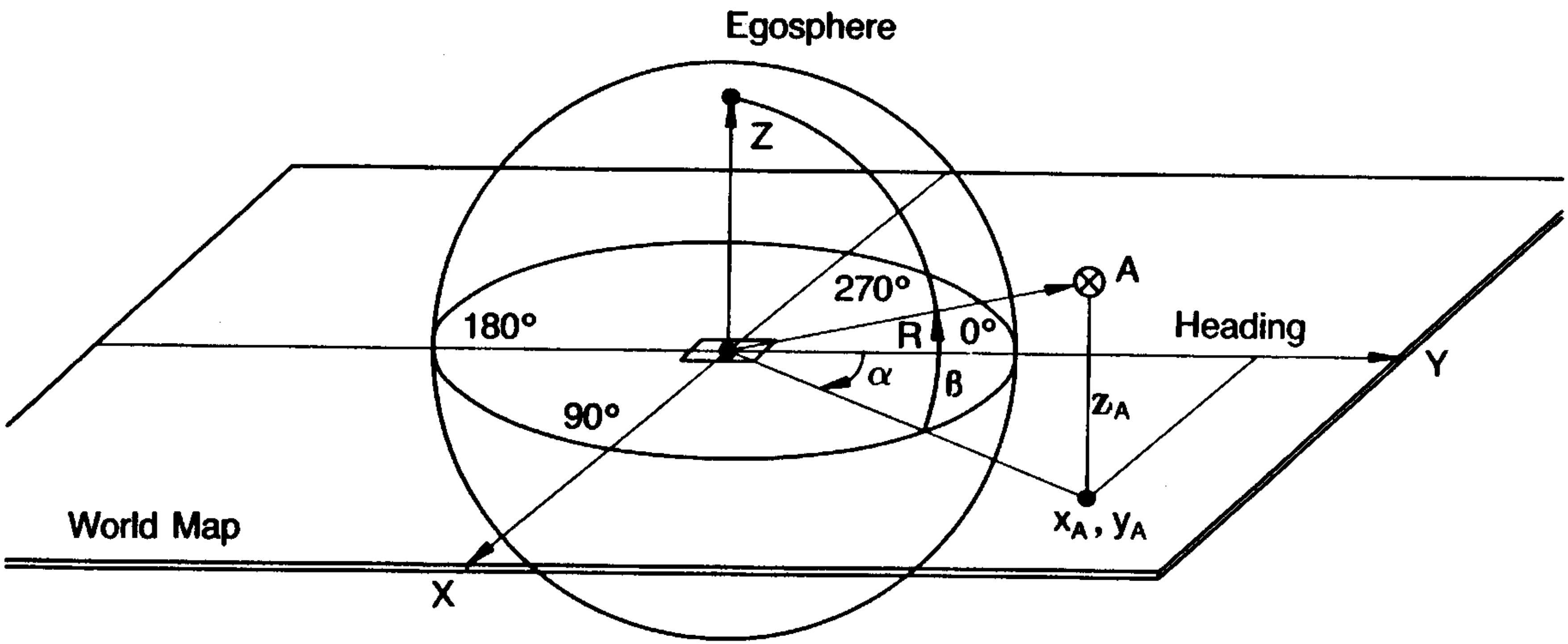


Figure 11: Three levels in a real-time planning hierarchy, illustrating the shrinking planning horizon and greater spatial and temporal detail at successively lower levels

vocabularies as the group and intergroup leaders of that individual, and those TD modules contain the same version of the command tree, and have the same information resident in their world models, and have the same set of goals, values, and priorities, then each individual can duplicate the computations performed by its group and intergroup leaders. Under such conditions, each individual can compute what commands it would receive if communications were available. It then can operate as if it were in constant communication, both with its superiors and peers, and cooperative behavior can take place without any communication at all. Even under conditions where task sequencing is event driven, if all relevant events can be sensed independently by all the cooperating individuals, task sequencing can be performed simultaneously by each individual, and communication is unnecessary.

Of course, such ideal conditions rarely obtain. Although, it is relatively simple and inexpensive to equip all individuals with the level 5 and 6 computing structure and algorithms, identical goals, priorities, values, and knowledge can seldom be long maintained for multiple independent individuals, and all relevant events will seldom be sensed and interpreted identically by all individuals. Communication thus becomes necessary for cooperative behavior. The role of communication is:

1) To issue commands and achieve a common goal
2) To define tasks and achieve a common strategy
3) To establish roles and achieve a common command tree
4) To communicate world states and achieve common world models.

The amount and frequency of communication required to achieve successful cooperative behavior is then inversely proportional to the amount of commonalty in goals, priorities, values, and knowledge.

REFERENCES

[1] Decker, K.S., "Distributed Problem-Solving Techniques: A Survey," IEEE Transactions on Systems, Man, and Cybernetics, Volume SMC-17, No. 5, September/October 1987.

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

[3] Fox, M.S., "An Organizational View of Distributed Systems," IEEE Transactions on Systems, Man, and Cybernetics, Volume SMC-11, pp. 70-80, January 1981.

[4] McLean, C.R. and Brown, P.E., "Process Planning in the AMRF," CIM Technology Magazine, August 1987.

[5] Cammarata, S., McArthur, D., and Steeb, R., "Strategies of Cooperation in Distributed Problem Solving," Proceedings 8th International Joint Conference on Artificial Intelligence, pp. 767-770, August 1983.

[6] Thorndyke, P.W., McArthur, D., and Cammarata, S., "Autopilot: A Distributed Planner for Air Fleet Control," Proceedings 7th International Joint Conference on Artificial Intelligence, Vancouver, BC, Canada, pp. 171-177, August 1981.

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

[8] Nilsson, N.J., "Two Heads are Better than One," SIGART Newsletter, p.43, October 1980.

[9] Minksy, M., <u>Society of Mind</u>, Simon and Schuster, New York, 1986.

[10] Lard, J.E., Newell, A., and Rosenbloom, P.S., "SOAR: An Architecture for General Intelligence," Artificial Intelligence 33 (1987), pp. 1-64, Elsevier Science Publishers B.V. (North Holland).

[11] Intelligent Task Automation Interim Technical Report, Report No. II-4, Honeywell Corporation Systems, Development Division, December 1987.

[12] Lowrie, J., et al., "Autonomous Land Vehicle," Annual Report, ETL-0413, Martin Marietta Denver Aerospace, July 1986.

[13] Albus, J.S., Barbera, A.J., and Nagel, R.N., "Theory and Practice of Hierarchical Control," Proceedings of the 23rd IEEE Computer Society Intenational Conference, September 1981.

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

[15] Leake, S. and Kilmer, R.D., "The NBS Real-Time Control System User's Reference Manual," NBS Technical Note 1258, June 1988.

[16] Shneier, M., Kent, E.W., Albus, J.S., Mansbach, P., Nashman, M., Palombo, M., Rutkowski, W., and Wheatley, T., "Robot Sensing for a Hierarchical Control System," Proceedings of the 13th ISIR/Robots 7 Symposium, Chicago, IL, April 1983.

[17] Kent, E.W. and Albus, J.S., "Servoed World Models as Interfaces between Robot Control Systems and Sensory Data," Robotica (1984) volume 2, pp. 17-25.

[18] Fiala J. and Wavering A., "An RCS Application Example: Tool Change on a Horizontal Machining Center," Proceedings of the Second International Conference on Robotics and Factories of the Future, San Diego, CA, July 28-31, 1987.

[19] Murphy, K.N., Norcross, R.J., and Proctor, F.M., "CAD Directed Robotic Deburring," Proceedings of the Second International Symposium on Robotics and Manufacturing Research, Education, and Applications, Alburquerque, NM, November 1988.

[20] McLean, C.R., "The Vertical Machining Workstation of the AMRF: Software Integration," Proceedings of the ASME Symposium on Integrated and Intelligent Manufacturing, Anaheim, CA, December 1986.

[21] Kramer, T.R., "Data Handling in the Vertical Workstation of the Automated Manufacturing Research Facility at the National Bureau of Standards," NSBIR 88-3763 (NB), April 1988.

[22] Hopp, T.H., "CAD-Directed Inspection," CIRP Annals, Volume 33/1, 1984.

[23] Mitchell, M.J. and Barkmeyer , E.J., "Data Distributed in the NBS Automated Manufacturing Research Facility," Proceedings of the IPAD2 Conference, Denver, CO, April 17-18, 1984.

[24] Furlani, C. (Editor), "Hierarchical Control System Emulation User's Manual," NBSIR 85-3156, April 1985.

[25] McCain, H.G., Kilmer, R.D., Szabo, S., and Abrishamian, A., "A Hierarchically Controlled Autonomous Robot for Heavy Payload Military Field Applications," Proceedings of the International Congress on Intelligent Autonomous Systems, Amsterdam, The Netherlands, December 8-11, 1986.

[26] Albus, J.S., McCain, H.G., and Lumia, R., "NASA/ NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)," NBS Technical Note 1235, July 1987.

[27] Fikes, R.E. and Nilsson, N.J., "Strips: A New Approach to the Application of Theorem Proving to Problem Solving," Artificial Intelligence, 2, (2), pp. 189-208, 1971.

[28] Sacerdoti, E., A Structure for Plans and Behavior, New York Elsevier North Holland, 1977.

[29] Schank, R.C. and Abelson, R.P., Scripts Plans Goals and Understanding, Hillsdale, NJ, Lawrence Erlbaum Associates, 1977.

[30] Arbib, M.A., The Metaphorical Brain, Wiley, New York, 1972.