The Role of World Modeling and Value Judgment in Perception

James S. Albus

Robot Systems Division
National Institute of Standards and Technology

Abstract

The world model is the intelligent system's best estimate of the external world. It provides an interface between sensory processing and task decomposition. A world model architecture is proposed that represents knowledge in terms of maps, entity frames, and state variables. This structure is hierarchically organized so as to provide multiple levels of resolution in space and time. Three types of map coordinates are important: egospheres, object coordinates, and world coordinates.

Value judgments provide an evaluation of hypothesized plans, and perceived objects, events, and situations. Evaluations produce value state-variables that indicate cost, benefit, risk, priority, desirability, attractiveness, and uncertainty.

Introduction

The world model is an intelligent system's internal representation of the external world. It is the system's best estimate of objective reality.

A clear distinction between an internal model of the world that exists in the mind, and the external world of reality where we all live and act, was first made by Schopenhauer about 100 years ago [1]. The concept of an internal world model is crucial to understanding perception and cognition. The world model is what provides the brain with knowledge of things that are not directly and immediately observable. The world model is what enables the mind to integrate noisy and intermittent sensory input from many different sources into a single reliable representation of spatio-temporal reality.

Knowledge in the world model may be represented either implicitly or explicitly. Implicit world knowledge may be embedded in the control and sensory processing algorithms and interconnections of a brain, or of a computer system. Explicit world knowledge may be represented in either natural or artificial systems by data in database structures such as maps, lists, and semantic nets. Explicit world models require computational modules capable of map transformations, indirect addressing, and list processing. Computer hardware and software techniques for implementing these types of functions are well known.

In the model presented here, explicit world model database structures are defined to reside in (or be accessed by) the global memory GM modules. Database management processes such as query processors, data servers, and question answering systems are defined to reside in the WM modules. Together the WM and GM modules make up the world model.

WM and GM modules

The world model is hierarchically structured and distributed such that there is a WM and GM module in each node at every level of the control hierarchy defined in [2]. At each level, the WM modules perform the functions illustrated in Figure 1.

1) WM modules maintain the GM knowledge database, keeping it current and consistent. In this role, the WM modules perform the functions of a database management system. They update GM state estimates based on correlations and differences between world model predictions and sensory observations at each hierarchical level. The WM modules enter newly recognized entities, states, and events into the GM database, and delete entities and states determined by the sensory processing modules to no longer exist in the external world. The WM modules also enter estimates, generated by the VJ modules, of the reliability of GM state variables. Believability or confidence factors are assigned to many types of state variables.

2) WM modules generate predictions of expected sensory input for use by the appropriate sensory processing SP modules. In this role, a WM module performs the functions of a graphics engine, or state predictor, generating predictions that enable the sensory processing system to perform correlation and predictive filtering. WM predictions are based on the state of the task and estimated states of the external world. For example in vision, a WM module may use the information in an object frame to generate predicted images which can be compared pixel by pixel, or entity by entity, with observed images.

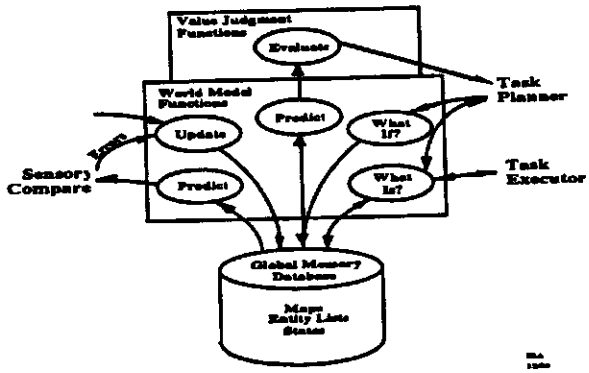

Figure 1. Functions performed by the WM module. 1) Update global memory with recognized entities. 2) Predict sensory data. 3) Answer "What is?" queries from task executor and return current state of world. 4) Answer "What if?" queries from task planner and predict results for evaluation.

Proceedings of the Fifth IEEE International Symposium on Intelligent Control, Philadelphia, PA, September 5-7, 1990

3) WM modules answer "What is?" questions asked by the planners and executors in the corresponding level TD modules. In this role, the WM modules perform the function of database query processors, or data servers. World model estimates of the current state of the world are used by TD module planners as a starting point for planning. Current state estimates are used by TD module executors for servoing and branching on conditions.
4) WM modules answer "What if?" questions asked by the planners in the corresponding level TD modules. In this role, the WM modules perform the function of simulation by generating expected results from actions hypothesized by the TD planners. Results predicted by WM simulations are sent to value judgment VJ modules for evaluation. For each hypothesized action, a VJ evaluation is returned to the TD planner. This TD, WM, VJ loop enables TD planners to select the sequence of hypothesized actions producing the best evaluation as the plan to be executed.

GM modules provide memory, communication, and switching services that make the world model behave like a global virtual common memory in response to queries and updates from the TD, WM, SP, and VJ modules. The GM module in each node provides a communication window (i.e. a network terminal, or mailbox interface) into the global virtual memory database for each of the TD, WM, SP, and VJ modules in that node.

Knowledge Representation

Knowledge in the world model database includes both a-priori information which is available to the intelligent system before action begins, and a-posterior knowledge which is gained from sensing the environment as action proceeds. World model knowledge includes information about space, time, entities, events, and states of the external world.

The world model also includes knowledge about the intelligent system itself, such as values assigned to motives, drives, and priorities; values assigned to goals, objects, and events; parameters embedded in kinematic and dynamic models of the limbs and body; states of internal pressure, temperature, clocks, and blood chemistry or fuel level; plus the states of all of the processes currently executing in each of the TD, SP, WM, and VJ modules.

Knowledge about space is represented in maps. Knowledge about entities, events, and states is represented in lists, or frames. Knowledge about the laws of physics, chemistry, optics, and the rules of logic and mathematics is represented in the WM functions that generate predictions and simulate results of hypothetical actions. Such knowledge may be represented as algorithms, or as IF/THEN rules of what happens under certain situations, such as when things are pushed, thrown, or dropped.

The correctness and consistency of world model knowledge is verified by sensory processing mechanisms that measure differences between world model predictions and sensory observations.

Space

From psychophysical evidence Gibson [3] concludes that the perception of space is primarily in terms of "medium, substance, and the surfaces that separate them". Medium is the air, water, fog, smoke, or falling snow through which the world is viewed. Substance is the material, such as earth, rock, wood, metal, flesh, grass, clouds, or water, that comprise the interior of objects. The surfaces that separate the viewing medium from the viewed objects is what are observed by the sensory system. The sensory input thus describes the external physical world primarily in terms of surfaces.

Surfaces are thus selected as the fundamental element for representing space in the proposed GM database. Volumes are treated as distances between surfaces. Objects are defined as circumscribed, often closed, surfaces. Lines, points and vertices lie on, and may define surfaces. Spatial relationships on surfaces are represented by maps.

Maps

Definition: A map is a two dimensional database that defines a mesh or grid on a surface.

The surface represented by a map may be, but need not be, flat. For example, a map may be defined on a surface that is draped over, or even wrapped around, a 3-dimensional volume.

Theorem: Maps can be used to describe the distribution of entities in space.

It is always possible and often useful to project the physical 3-D world onto a 2-D surface defined by a map. For example, most commonly used maps are produced by projecting the world onto the 2-D surface of a flat sheet of paper, or the surface of a globe. One great advantage of such a projection is that it reduces the dimensionality of the world from three to two. This produces an enormous saving in the amount of memory required for a database representing space.

Map Overlays

Most of the useful information lost in the projection from 3-D space to a 2-D surface can be preserved through the use of map overlays.

Definition: A map overlay is an assignment of values, or parameters, to points on the map.

A map overlay can represent spatial relationships between 3-D objects. For example, an object overlay may indicate the presence of buildings, roads, bridges, and landmarks at various places on the map. Objects that appear smaller than a pixel on a map can be represented as icons. Larger objects may be represented by labeled regions that are projections of the 3-D objects on the 2-D map. Objects appearing on the map overlay may be cross referenced to an object database frame elsewhere in the world model. Information about the 3-D geometry of objects on the map may be represented in the object frame database.

Map overlays can also indicate attributes associated with points (or pixels) on the map. One of the most common map overlays defines terrain elevation. A value of terrain elevation (z) overlaid at each (x,y) point on a world map produces a topographic map.

A map can have any number of overlays. Map overlays may indicate brightness, color, temperature, etc. A brightness or color overlay may correspond to a visual image. For example, when aerial photos or satellite images

are registered with map coordinates, they become brightness or color map overlays.

Map overlays may indicate terrain type, or region names, or can indicate values, such as cost or risk, associated with regions. Map overlays can indicate which points on the ground are visible from a given location in space. Overlays may also indicate contour lines and grid lines such as latitude and longitude, or range and bearing.

Map overlays may be useful for a variety of functions. For example, terrain elevation and other characteristics may be useful for route planning, or for planning and executing tasks of manipulation and locomotion. Object overlays can be useful for analyzing scenes and recognizing objects and places.

A map typically represents the configuration of the world at a single instant in time, i.e. a snapshot. Motion can be represented by overlays of state variables such as velocity or image flow vectors, or traces (i.e. trajectories), of entity locations. Time may be represented explicitly by a numerical parameter associated with each trajectory point, or implicitly by causing trajectory points to fade, or be deleted, as time passes.

Theorem: A set of map overlays are equivalent to a set of map pixel frames.

If each map overlay defines a parameter value for each map pixel, then the set of all overlay parameter values for each map pixel defines a frame for that pixel. The frame for each pixel thus describes the region covered by that pixel. For example, a pixel frame may describe the color, range, and orientation of the surface covered by the pixel. It may also describe the name of the object to which the surface covered by the pixel belongs, and the value state-variables assigned to the object.

Map resolution

The resolution required for a world model map depends on how the map is generated and how it is used.

For predicting sensory input, world model maps need to have resolution comparable to the resolution of the sensory system. For vision, map resolution may be on the order of a hundred thousand to a million pixels or more. For other sensory modalities, it can be considerably less.

For planning, different levels of the control hierarchy require maps of different scale. At higher levels, plans cover long distances and times, and require maps of large area, but low resolution. At lower levels, plans cover short distances and times, and maps need to cover small areas with high resolution.

For long term memory, world model maps can have resolution on the order of a few thousand pixels or less. For example, few humans can recall from memory the spatial distribution of as many as a hundred objects, even in familiar locations such as their own homes.

The spatial memory of an intelligent creature typically consists of a finite number of relatively small regions that may be widely separated in space; for example, one's own home, the office, or school, the homes of friends and relatives, etc. These known regions are typically connected by pathways that contain at most a few hundred known waypoints and branchpoints. The remainder of the world is known little, or not at all. Unknown regions, which make up the vast majority of the real world, occupy little or no space in the world model.

The efficient storage of maps with extremely non-uniform resolution can be accomplished in a computer database by quadtree [4], hash coding, or other sparse memory representations [5]. Pathways between known areas can be economically represented by graph structures either in neuronal or electronic memories. Neural net representations [6] give insight as to how non-uniformly dense spatial information might be represented in the brain.

Maps and Egospheres

There are three general types of map coordinate frames that are important to an intelligent system: world coordinates, object coordinates, and egospheres.

World coordinates

World coordinates are often expressed in a Cartesian frame, and referenced to a point in the world. In most cases, the origin is an arbitrary point on the ground. The z axis is defined by the vertical, and the x and y axes define points on the horizon. For example, y may point North and x East. The value of z is often set to zero at sea level.

World coordinates may also be referenced to a moving point in the world. For example, the origin may be the self, or some moving object in the world. In this case, stationary pixels on the world map must be scrolled as the reference point moves.

There may be several world maps with different resolutions and ranges.

Object coordinates

Object coordinates are defined with respect to features of an object. For example, the origin might be defined as the center of gravity, with the coordinate axes defined by axes of symmetry, faces, edges, vertices, or skeletons [7]. There are a variety of surface representations that have been suggested for representing object geometry. Among these are generalized cylinders [8,9], B-splines [10], quadtrees [4], and aspect graphs [11].

The tight coupling between sensory processing and world modeling suggests that intelligent biological creatures use a representation that: 1) can be updated on a pixel by pixel basis, and 2) can produce predicted images and map overlays in real-time with resolution comparable to sensory input.

Egospheres

An egosphere is a 2-dimensional spherical surface that is a map of the world as seen by an observer at the center of the sphere. Visible points on regions or objects in the world are projected on the egosphere wherever the line of sight from a sensor at the center of the egosphere to the points in the world intersect the surface of the sphere. Egosphere coordinates thus are polar coordinates defined by the self at the origin. As the self moves, the projection of the world flows across the surface of the egosphere.

Just as the world map is a flat 2-D (x,y) array with multiple overlays, so the egosphere is a spherical 2-D (AZ,EL) array with multiple overlays. Egosphere overlays can attribute brightness, color, range, image flow, texture, and other properties to regions and entities on the egosphere. Regions on the egosphere can thus be segmented by attributes, and egosphere points with the same attribute value may be connected by contour lines. Egosphere overlays may also indicate the trace, or history, of brightness values or entity positions over some time interval. Objects may be represented on the egosphere by icons, and each object may have in its database frame a trace, or trajectory, of positions on the egosphere over some time interval.

Map transformations

Theorem: If surfaces in real world space can be covered by an array (or map) of points in a coordinate system defined in the world, and the surface of the WM egosphere is also represented as an array of points, then there exists a function G that transforms each point on the real world map into a point on the WM egosphere, and a function G' that transforms each point on the WM egosphere into a point on the real world map.

For example, Figure 2 shows the 3-D relationship between an egosphere and world map coordinates. For every point (x,y,z) in world coordinates, there is a point (AZ,EL,R) in ego centered coordinates which can be computed by the 3x3 matrix function G

$$(AZ,EL,R)^T = G\ (x,y,z)^T$$

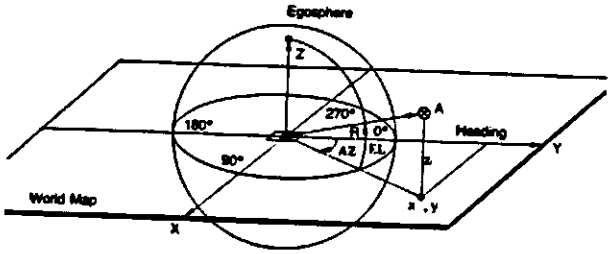

Figure 2. Geometric relationship between world map and egosphere coordinates.

There, of course, may be more than one point in the world map that gives the same (AZ,EL) values on the egosphere. Only the (AZ,EL) with the smallest value of R will be visible to an observer at the center of the egosphere. The deletion of egosphere pixels with R larger than the smallest for each value of (AZ,EL) corresponds to the hidden surface removal problem common in computer graphics.

For each egosphere pixel where R is known, (x,y,z) can be computed from (AZ,EL,R) by the function G'

$$(x,y,z)^T = G'\ (AZ,EL,R)^T$$

Any point in the world topological map can thus be projected onto the egosphere (and vice versa when R is known). Projections from the egosphere to the world map will leave blank those map pixels that cannot be observed from the center of the egosphere.

There are 2x2 transformations of the form

$$(AZ,EL)^T = F\ (az,el)^T$$

and

$$(az,el)^T = F'\ (AZ,EL)^T$$

that can relate any map point (AZ,EL) on one egosphere to a map point (az,el) on another egosphere.The radius R to any egosphere pixel is unchanged by the F and F' transformations, since all egosphere representations have the same origin.

As ego motion occurs (i.e. as the self object moves through the world), the egosphere moves relative to world coordinates, and points on the egocentric maps flow across their surfaces. Ego motion may involve translation, or rotation, or both; in a stationary world, or a world containing moving objects. Once range to a stationary point in the world is known, its pixel motion on the egosphere can be predicted from knowledge of egomotion. For moving points, prediction of pixel motion on the egosphere requires additional knowledge of object motion.

Egosphere Representations

The world model contains at least four egosphere representations:

1) Sensor Egosphere

The sensor egosphere is defined by the sensor position and orientation, and moves as the sensor moves. For vision, the sensor egosphere is the coordinate system of the retina. The sensor egosphere has coordinates of azimuth (AZ) and elevation (EL) fixed in the sensor system (such as the eye or a TV camera), as shown in Figure 3. For a narrow field of view, rows and columns (x,z) in a flat camera image array correspond quite closely to azimuth and elevation (AZ,EL) on the sensor egosphere. However, for a wide field of view, the egosphere and flat image array representations have widely different geometries. The flat image (x,z) representation becomes highly elongated for a wide field of view, going to infinity at plus and minus 90 degrees. The egosphere representation, in contrast, is well behaved over the entire sphere.

The sensor egosphere representation is useful for the analysis of wide angle vision such as occurs in the eyes of most biological creatures. For example, most insects and fish, many birds, and most prey animals such a rabbits have eyes with fields of view of 180 degrees or more. Such eyes are often positioned on opposite sides of the head so as to provide almost 360 degree visual coverage. The sensor egosphere representation provides a tractable coordinate frame in which this type of vision can be analyzed.

2) Head Egosphere

The head egosphere has (az,el) coordinates measured in a reference frame fixed in the head (or sensor platform). The head egosphere representation is well suited for fusing sensory data from multiple sensors, each of which has its own coordinate system. Vision data from multiple eyes or cameras can be overlaid and registered in order to compute range from stereo. Directional and range data from acoustic and sonar sensors can be overlaid on vision data. Data derived from different sensors, or from multiple readings of a single moving sensor, can be overlaid on the head egosphere to build up an image.

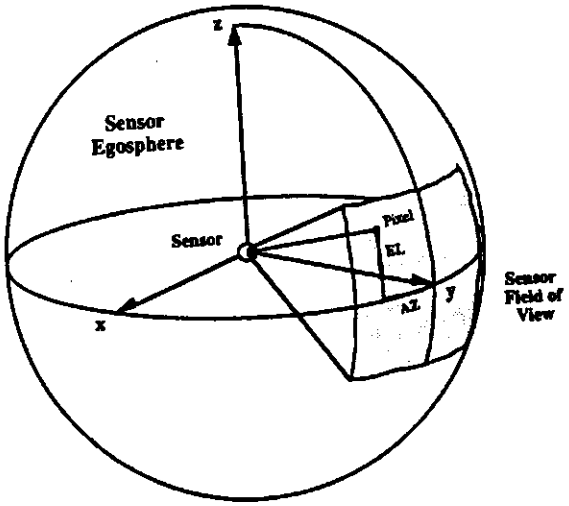

Figure 3. Sensor egosphere coordinates. Azimuth (AZ) is measured clockwise from the sensor y-axis in the x-y plane. Elevation is measured up and down (plus and minus) from the x-y plane.

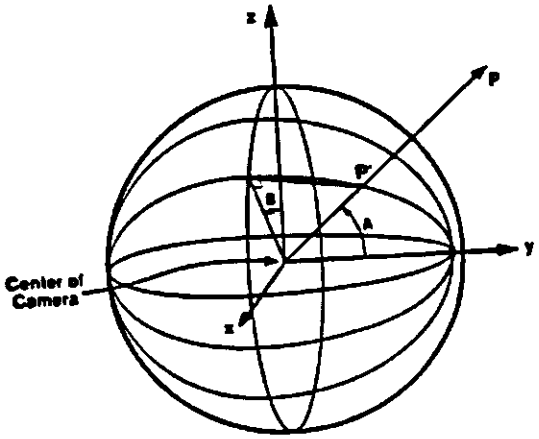

Figure 4. The velocity egosphere. On the velocity egosphere, the y-axis is defined by the velocity vector. The x-axis points to the horizon on the right. A is the angle between the velocity vector and a pixel on the egosphere, and B is the angle between the z-axis and the plane defined by the velocity vector and the pixel vector.

3) Velocity Egoshpere

The velocity egosphere is defined by the velocity vector and the horizon. The velocity vector defines the pole (y-axis) of the velocity egosphere, and the x-axis points to the horizon as shown in Figure 4. The egosphere coordinates (A,B) are defined such that A is the angle between the pole and a pixel, and B is the angle between the y-o-z plane and the plane of the polar great circle containing the pixel.

For egocenter translation without rotation through a stationary world, image flow occurs entirely along great circle arcs defined by B = constant. The positive y-axis of the velocity egosphere thus corresponds to the focus-of-expansion. The negative y-axis corresponds to the focus-of-contraction. The velocity egosphere is ideally suited for computing range from image flow. The range for each pixel is given by

$$R = \frac{v \sin A}{dA/dt}$$

where R is the range to the point
v is translational velocity of the eye
A is the angle between the velocity vector and the pixel covering the point
dA/dt is the image flow rate at the pixel covering the point

The computation of R is thus one dimensional in A, and depends only on the velocity of the eye and the flow rate along the B=constant circle through the pixel.

4) Inertial Egosphere

The inertial egosphere has coordinates of azimuth measured from a fixed point (such as North) on the horizon, and elevation measured from the horizon.

The inertial egosphere does not rotate with respect to distant objects and gravity as a result of sensor or body motion. On the inertial egosphere, the world is perceived not to rotate despite rotary motion of the sensors and the head.

Figure 5 illustrates the relationships between the various egosphere representations. Pixel data in eye (or camera) egosphere coordinates can be transformed into head (or sensor platform) egosphere coordinates by knowledge of the position and orientation of the sensor relative to the head. For example, the position of each eye in the head is fixed and the orientation of each eye relative to the head is known from stretch receptors in the ocular muscles (or pan and tilt encoders on a camera platform). Pixel data in head egosphere coordinates can be transformed into inertial egosphere coordinates by knowing the orientation of the head in inertial space. This information can be obtained from the vestibular (or inertial) system that measures the direction of gravity relative to the head and integrates rotary accelerations to obtain head position in inertial space. The inertial egosphere can be transformed into world coordinates by knowing the x,y,z position of the center of the egosphere. This is obtained from knowledge about where the self is located in the world. Pixels on any egosphere can be transformed into the velocity egosphere by knowledge of the direction of the current velocity vector on that egosphere. This can be obtained from a number of sources including the locomotion and vestibular systems.

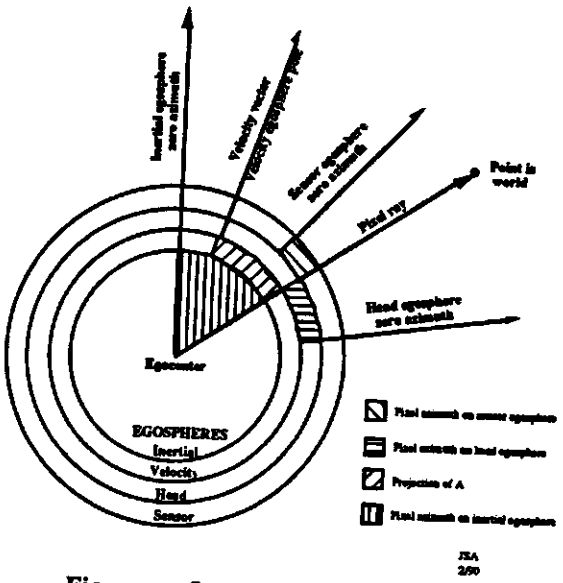

Figure 5. A 2-D projection of four egosphere representations illustrating angular relationships between egospheres. Pixels are represented on each egosphere such that images remain in registration. Pixel attributes detected on one egosphere may thus be inherited on others. Pixel resolution is not typically uniform on a single egosphere, nor is it necessarily the same for different egospheres, or for different attributes on the same egosphere.

Transformations to and from the sensor egosphere, the inertial egosphere, the velocity egosphere, and the world map allow the intelligent system to sense the world from one perspective and interpret it in another. They allow the intelligent system to compute how entities in the world would look from another viewpoint. They provide the ability to overlay sensory input with world model predictions, and to compute the geometrical and dynamical functions necessary to navigate, focus attention, and direct action relative to entities and regions of the world.

All of the above egosphere transformations can be inverted, so that conversions can be made in either direction. Each transformation consists of a relatively simple vector function computed on each pixel. Full image egosphere transformations can be accomplished at television frame rates by state-of-the-art serial computing hardware. They can be accomplished in microseconds by parallel hardware.

Entities

Definition: An entity is an element from the set {point, line, surface, object, group}

The world model contains information about entities stored in lists, or frames. The GM database contains a list of all the entities that the intelligent system knows about. A subset of this list is the set of current entities known to be present in any given situation. A subset of the list of current entities is the set of entities of attention.

There are two types of entities: generic and specific. A generic entity is an example of a class of entities. A generic entity frame contains the attributes of its class. A specific entity is a particular instance of an entity. A specific entity frame inherits the attributes of the class to which it belongs.

An example of an entity frame is:

ENTITY NAME	-- name of entity
kind	-- class or species of entity
type	-- generic or specific point, line, surface, object, or group
position	-- world map coordinates (uncertainty) egosphere coordinates (uncertainty)
dynamics	-- velocity (uncertainty) acceleration (uncertainty)
trajectory	-- sequence of positions
geometry	-- axis of symmetry (uncertainty) size (uncertainty) shape
links	-- subentities parent entity
properties	-- physical mass color substance behavioral social (of animate objects)
capabilities	-- speed, range
value state-variables	-- attract-repulse confidence-fear love-hate

For example, upon observing a specific cow named Bertha, an entity frame in the brain of a visitor to a farm might have the following values:

ENTITY NAME	-- Bertha
kind	-- cow
type	-- specific object
position	-- x,y,z (in pasture map coordinates) -- AZ, EL, R (in egosphere image of observer)
dynamics	-- velocity = 0 acceleration = 0
trajectory	-- sequence of positions while grazing
geometry	-- axis of symmetry (right/left) size (6x3x10 ft) shape (quadruped)
links	-- subentities - surfaces (torso, neck, head, legs, tail, etc.)
	-- parent entity - group (herd)
properties	-- physical mass (1050 lbs) color (black and white) substance (flesh, bone, skin, hair)
	-- behavioral (standing, placid, timid, etc.)
capabilities	-- speed, range
value state-variables	-- attract-repulse = 3 (visitor finds cows attractive to look at confidence-fear = -2 (visitor slightly afraid of cows) love-hate = 1 (no strong feelings)

Map - Entity Relationship

Map and entity representations must be cross referenced and tightly coupled by real-time computing hardware. Each pixel on the map can have in its frame a pointer to the list of entities covered by that pixel. For example, each pixel may cover a point entity indicating brightness, color, spatial and temporal gradients of brightness and color, image flow, and range for each point. Each pixel may also cover a linear entity indicating a brightness or depth edge or vertex; a surface entity indicating area, slope, and texture; an object entity indicating the name and attributes of the object covered; a group entity indicating the name and attributes of the group covered, etc.

Likewise, each entity in the attention list may have in its frame a set of geometrical parameters that enables the geometry engine to compute the set of egosphere or world map pixels covered by each entity, so that entity parameters associated with each pixel can be overlaid on the world and egosphere maps.

Such cross referencing between pixel maps and entity frames allows the results of each level of processing to add map overlays to the egosphere and world map representations. The entity database can be updated from knowledge of image parameters at points on the egosphere, and the map database can be predicted from knowledge of entity parameters in the world model. At each level, local entity and map parameters can be computed in parallel.

Many of the attributes in an entity frame may be time dependent state-variables. Each time dependent state-variable may possess its own short term memory queue wherein is stored a state trajectory, or trace, that describes its temporal history.
At each hierarchical level, temporal traces stretch backward about as far as the planning horizon at that level stretches into the future. At each hierarchical level, the historical trace of an entity state-variable may be captured by summarizing data values at several points in time throughout the historical interval. Time dependent entity state-variable histories may also be captured by running averages and moments, Fourier transform coefficients, Kalman filter parameters, or other analogous parameters.

Each state-variable in an entity frame may have value state-variable parameters that indicate levels of believability, confidence, support, or plausibility, and measures of dimensional uncertainty. These are computed by value judgment functions that reside in the VJ modules.

Value state-variable parameters may be overlaid on the map and egosphere regions where the entities to which they are assigned appear. This facilitates planning. For example, approach-avoidance behavior can be planned on an egosphere map overlay defined by the summation of attractor and repulsor value state-variables assigned to objects or regions that appear on the egosphere. Navigation planning can be done on a map overlay whereon risk and benefit values are assigned to regions on the egosphere or world map.[12]

Entity Database Hierarchy

The entity database is hierarchically structured. Each entity consists of a set of subentities, and is part of a parent entity. For example, an object may consist of a set of surfaces, and be part of a group.

The definition of an object is quite arbitrary, however, at least from the point of view of the world model. For example, is a nose an object? If so, what is a face? Is a head an object? Or is it part of a group of objects comprising a body? If a body can be a group, what is a group of bodies?

Only in the context of a task, does the definition of an object become clear. For example, in a task frame, an object may be defined either as the agent, or as acted upon by the agent executing the task. Thus, in the context of a specific task, the nose (or face, or hand) may become an object because it appears in a task frame as the agent or object of a task.

Thus, perception in an intelligent system is task (or goal) driven, and the structure of the world model entity database is defined by, and may be reconfigured by, the nature of goals and tasks. It is, therefore, not necessarily the role of the world model to define the boundaries of entities, but rather to represent the boundaries defined by the task frame, and to map regions and entities circumscribed by those boundaries with sufficient resolution to accomplish the task. It is the role of the sensory processing system to identify regions and entities in the external real world that correspond to those represented in the world model, and to discover boundaries that circumscribe objects defined by tasks.

This suggests that the hierarchical structure of the world model map and entity database might be placed in one-to-one correspondence with the hierarchical levels of task decomposition proposed in [2]. For example at level 1 of the task decomposition hierarchy, the world model should represent point entities in head egosphere coordinates. In the case of vision, point entities may consist of brightness or color intensities, spatial and temporal derivatives of those intensities, image flow vectors, and range estimates for each pixel.

At level 2 of the task decomposition hierarchy, the world model should represent linear entities consisting of clusters of point entities. In the visual system, linear entities consist of brightness or color edges, depth edges, vertices, or trajectories of points. Attributes such as position, orientation, and velocity are represented for each linear entity. Linear entities are represented in inertial (or possibly velocity) egosphere coordinates. These representations are roughly analogous to Marr's "primal sketch". [13] This corresponds to data representations in the primary visual cortex.

At level 3, the world model should represent surface entities consisting of sets of linear entities clustered or swept into bounded surfaces or maps, such as terrain maps, B-spline surfaces, or general two dimensional functions. Surface entities are represented in object coordinates. In the case of vision, entity attributes may describe surface orientation, surface velocity, range, and surface discontinuities or boundaries. Level 3 thus corresponds to data representation in the secondary visual cortex, or Marr's "2 1/2-D sketch".

At level 4, object entities should consist of sets of surfaces clustered or swept so as to define 3-D volumes, or objects in world map coordinates. World map resolution may be on the order of feet, and range on the order of hundreds of feet. In the case of vision, object entities may include occluding

contours, axes of symmetry, volumes, etc. This corresponds to data representations in visual association areas of the parietal cortex. These are analogous to Marr's "3-D model" representation.

At level 5, group entities should consist of sets of objects that are clustered into groups or packs. Group entities are represented in world map coordinates. Resolution may be on the order of tens of feet, and range on the order of a mile.

At level 6, sets of groups may be clustered into groups of groups, or group^2 entities. Maps are in world coordinates with less resolution and more range than at level 5.

At level 7, sets of group^2 entities may be clustered into group^3 (or world) entities, and so on. At each higher level, world map resolution decreases and range increases by about an order of magnitude per level.

The highest level entity in the world model is the world itself, i.e. the environment as a whole. The environment entity frame contains attribute state-variables that describe the state of the environment, such as temperature, wind, precipitation, illumination, visibility, the state of hostilities or peace, etc.

Events

Definition: An event is a state, condition, or situation that exists at a point in time, or occurs over an interval in time.

Events are represented in the world model by frames that include the point, or interval, in time when the event occurred, or is expected to occur. Event frames also contain lists of attributes such as start and end time, duration, type, relationship to other events, etc.

An example of an event frame is:

EVENT NAME	-- name of event
kind	-- class or species
type	-- generic or specific
modality	-- visual, auditory, tactile, etc.
time	-- when event detected
interval	-- period over which event took place
position	-- map location where event occurred
links	-- subevents
	-- parent event
value	-- good-bad, benefit-cost, etc.

A sequence of subevents may be represented by a string of symbols, where each symbol corresponds to the name of an event.

The event entity database is hierarchical, and the hierarchical levels can be placed in one-to-one correspondence with levels in the task decomposition hierarchy defined in [2]. At each level, an event may consist of the recognition of a pattern, or string, of subevents. For example at:

Level 1 -- an event may span a few milliseconds. A typical level one event might be the recognition of a tone, hiss, click, a change in pixel intensity, or a measurement of image flow at a pixel.

Level 2 -- an event may span a few tenths of a second. A typical level two event might be the recognition of a phoneme, a musical chord, or trajectory of a visual point or feature.

Level 3 -- an event may span a few seconds, and consist of the recognition of a word, a short phrase, a visual gesture, or motion of a visual surface.

Level 4 -- an event may span a few tens of seconds, and consist of the recognition of a message, a melody, a visual observation of object motion, or task activity.

Level 5 -- an event may span a few minutes and consist of listening to a conversation, a song, or visual observation of group activity in an extended social exchange.

Level 6 -- an event may span an hour and include many auditory, tactile, and visual observations.

Level 7 -- an event may span a day and include sensory observations of an entire day's activities.

State-variables in the event frame may have confidence levels, degrees of support and plausibility, and measures of dimensional uncertainty similar to those in spatial entity frames. Additional confidence state-variables may indicate the degree of certainty that an event actually occurred, or was correctly recognized.

Value Judgments

Value judgments provide the criteria for making intelligent choices. Value judgments evaluate the costs, risks, and benefits of plans and actions, and the desirability, attractiveness, and uncertainty of objects and events. Value judgment modules produce evaluations that can be represented as value state-variables. These can be assigned to the attribute lists in entity frames of objects, persons, events, situations, and regions of space. They can also be assigned to the attribute lists of plans and actions in task frames. Value state-variables can therefore label entities, actions, and plans as good or bad, as important or trivial, as desirable or undesirable. This information is used by the goal selection and task decomposition modules both for planning and executing actions. It provides the criteria for decisions about which coarse of action to take.

The Limbic System

In animal brains, value judgment functions are computed by the limbic system. Value state-variables produced by the limbic system include emotions, drives, and priorities. In animals and humans, electrical or chemical stimulation of specific limbic regions (i.e. value judgment modules) has been shown to produce pleasure and pain as well as more complex emotional feelings such as fear, anger, joy, contentment, and despair [14]. Other regions of the limbic system are responsible for computing levels of hunger, thirst, and sexual arousal. Others compute the body's level of arousal in response to danger and stress. Still others generate state-variables that indicate what is important and should be remembered, and what is unimportant and can safely be forgotten.

Input to and output from the limbic system connects to sources of highly processed sensory data, and to high level goal selection centers. Connections with the frontal cortex suggests that the value judgment modules are intimately involved with long range planning and geometrical

reasoning. Connections with the thalamus suggests that the limbic value judgment modules have access to high level perceptions about objects, events, relationships, and situations; for example, the recognition of success in goal achievement, the perception of praise or hostility, or the recognition of gestures of dominance or submission. Connections with the reticular formation suggests that the limbic VJ modules are also involved in computing confidence factors derived from the degree of correlation between predicted and observed sensory input. A high degree of correlation produces emotional feelings of confidence. Low correlation between predictions and observations generates feelings of fear and uncertainty. Input from the rhinencephelon conveys information about odor and taste.

It has long been recognized by psychologists that emotions play a central role in behavior. Fear leads to flight, hate to rage and attack. Joy produces smiles and dancing. Despair produces withdrawal and despondent demeanor. All creatures tend to repeat what makes them feel good, and avoid what they dislike. All attempt to prolong, intensify, or repeat those activities that give pleasure or make the self feel confident, joyful, or happy. All try to terminate, diminish, or avoid those activities that cause pain, or arouse fear, or revulsion.

It is common experience that emotions provide an evaluation of the state of the world as perceived by the sensory system. Emotions tell us what is good or bad, what is attractive or repulsive, what is beautiful or ugly, what is loved or hated, what provokes laughter or anger, what smells sweet or rotten, what feels pleasurable, and what hurts.

It is also widely known that emotions affect memory. Emotionally traumatic experiences are remembered in vivid detail for years, while emotionally non-stimulating everyday sights and sounds are forgotten within minutes after they are experienced.

Yet, emotions are popularly believed to be something apart from intelligence -- irrational, beyond reason or mathematical analysis. The theory presented here maintains the opposite. It treats emotion as a critical component of biological intelligence, necessary for evaluating sensory input, selecting goals, directing behavior, and controlling learning.

In the model proposed here, emotions are defined as value state-variables produced by the value judgment system. They may be assigned to frames as attributes that label objects, events, situations, and plans as good or bad, attractive or repulsive, desirable or undesirable. Objects or regions labeled with fear will be avoided. Those labeled with love will be pursued and protected. Those labeled with hate will be attacked, etc.

Priorities are value state-variables that provide estimates of importance. They can be assigned to task frames so that TD planners and executors can decide what to do first, how much effort to spend, how much risk is prudent, and how much cost is acceptable, for each task.

Drives are value state-variables that provide estimates of need. They can be assigned to the self frame, to indicate internal system needs and requirements. In biological systems, drives indicate levels of hunger, thirst, and sexual arousal. In mechanical systems, drives might indicate how much fuel is left, how much pressure is in a boiler, how many expendables have been consumed, or how much battery charge is remaining.

It is widely believed that machines cannot experience emotion, or that it would be dangerous, or even morally wrong to attempt to endow machines with emotions. However, unless machines have the capacity to make value judgments (i.e. to evaluate costs, risks, and benefits, to decide which coarse of action, and what expected results, are good, and which are bad) machines can never be intelligent or autonomous.

Without value judgments, machines cannot be intelligent, for they would have no basis for deciding to do one thing and not another, even to turn right rather than left. Without value judgments to support decision making, nothing can be intelligent, be it a machine or a biological creature.

VJ modules

In the model proposed here, value state-variables are computed by value judgment functions residing in VJ modules. Inputs to VJ modules describe entities, events, situations, and states. VJ value judgment functions compute measures of cost, risk, and benefit. VJ outputs are value state-variables. The VJ value judgment mechanism can thus be defined as a mathematical or logical function of the form

$$E = V(S)$$

where

E is an output vector of value state-variables

V is a value judgment function that computes E given S

S is an input state vector defining conditions in the world model, including the self. The components of S are entity attributes describing states of tasks, objects, events, or regions of space. These may be derived either from processed sensory information, or from the world model.

The value judgment function V in the VJ module computes a numerical scalar value (i.e. an evaluation) for each component of E as a function of the input state vector S. E is a time dependent vector. The components of E may be assigned to attributes in the world model frame of various entities, events, or states.

If time dependency is included, the function $E(t+dt) = V(S(t))$ may be computed by a set of equations of the form

$$e(j,t+dt) = (k\, d/dt + 1) \sum_i s(i,t)\, w(i,j)$$

where

e(j,t) is the value of the j-th value state-variable in the vector E at time t

s(i,t) is the value of the i-th input variable at time t

w(i,j) is a coefficient, or weight, that defines the contribution of s(i) to e(j).

Each individual may have a different set of "values", i.e. a different weight matrix in its value judgment function V.

The factor (k d/dt + 1) indicates that a value judgment may be dependent on the temporal derivative of its input variables as well as on their steady-state values. If $k > 1$, then the rate of change of the input factors become more important than their absolute values. Because of this k factor, need reduction and escape from pain are rewarding. The more rapid the escape, the more intense, but short-lived, the reward. Also because of this k factor, hope reduction and expectation downgrading are punishing.

Value State-Variable Map Overlays

When objects or regions of space are projected on a world map or egosphere, the value state-variables in the frames of those objects or regions can be represented as overlays on the projected regions. When this is done, value state-variables such as comfort, fear, love, hate, danger, and safe will appear overlaid on specific objects or regions of space. TD modules can then perform path planning algorithms that steer away from objects or regions overlaid with fear, or danger, and steer toward or remain close to those overlaid with attractiveness, or comfort. Task decomposition may generate attack commands for target objects or persons overlaid with hate. Protect, or care-for, commands may be generated for target objects overlaid with love.

Projection of uncertainty, believability, and importance value state-variables on the egosphere enables TD modules to perform the computations necessary for manipulating sensors and focusing attention.

Confidence, uncertainty, and hope state-variables may also be used to modify the effect of other value judgments. For example, if a task goal frame has a high hope variable but low confidence variable, behavior may be directed toward the hoped-for goal, but cautiously. On the other hand, if both hope and confidence are high, pursuit of the goal may be much more vigorous and intense.

The real-time computation of value state-variables varying task and world model conditions provides the basis for complex situation dependent behavior.

Summary and Conclusions

World modeling and value judgments are crucial to intelligent machine systems. The world model supports the planning and execution functions of task decomposition as described in [2], as well as the perceptual functions of sensory processing as described in [15]. In the world model, spatial knowledge is represented in an egocentric coordinate system in the form of maps and map overlays. Symbolic knowledge is represented in an object oriented structure, and the iconic and symbolic forms are cross referenced. Value judgments provide the criteria for decision making.

REFERENCES

1. Schopenhauer, A. *The World As Will and Idea, 1883* , In *The Philosophy of Schopenhauer*, Edited by Irwin Edman, Random House, New York, NY, 1928

2. Albus, J.S., "A Theory of Intelligent Systems, Proceedings IEEE Conference on Intelligent Control", Sept 1990, Philadelphia, PA

3. Gibson, J.J. *The Ecological Approach to Visual Perception*, Cornell University Press, Ithaca, N.Y. 1966

4. Samet, H., "The Quadtree and Related Hierarchical Data Structures", Computer Surveys, 16-2, 1984

5. Kinerva, P., *Sparse Distributed Memory*, MIT Press, Cambridge 1988.

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

7. Bradey, M., "Computational approaches to image understanding", ACM Computing Surveys 14, March, 1982

8. Binford, T. , "Inferring surfaces from images", Artif. Intell 17, 205-244, 1981

9. Marr, D., and H.K. Nishihara. "Representation and recognition of the spatial organization of three-dimensional shapes. Proc, R. Soc, Lond. B 200, 269-294, 1978

10. Riesenfeld,R.F.,"Application of B-spline Approximation to Geometric Problems of Computer Aided Design", Ph.D Thesis, Syracuse University (1973). Available at University of Utah, UTEC -CSc-73-126.

11. Koenderink, J.J., "The Structure of Images", Biological Cybernetics, 50, 1984.

12. Payton, D. W., "Internalized Plans: A Representation for Action Resources", Robotics and Autonomous Systems, 6, 89-103, 1990

13. Marr, D., *Vision*, W.H. Freeman, San Francisco, 1982

14. Guyton, A.C., *Organ Physiology, Structure and Function of the Nervous System*, 2nd ed., Philadelphia: W.B. Saunders, 1976

15. Albus, J.S., "Hierarchical Interaction Between Sensory Processing and World Modeling in Intelligent Systems", Proceedings IEEE Conference on Intelligent Control, Sept 1990, Philadelphia, PA