An Intelligent Systems Architecture for Manufacturing

James S. Albus

Intelligent Systems Division
National Institute of Standards and Technology
albus@cme.nist.gov

Abstract

The Intelligent Systems Architecture for Manufacturing (ISAM) is a reference model architecture for intelligent manufacturing systems. It is intended to provide a theoretical framework for the development of standards and performance measures for intelligent manufacturing systems. It also intended to provide engineering guidelines for the design and implementation of intelligent control systems for a wide variety of manufacturing applications.

ISAM consists of a hierarchically layered set of intelligent processing nodes organized as a nested series of control loops. In each node, tasks are decomposed, plans are generated, world models are maintained, feedback from sensors is processed, and control loops are closed. In each layer, nodes have a characteristic span of control, with a characteristic planning horizon, and corresponding level of detail in space and time. Nodes at the higher levels deal with corporate and production management, while nodes at lower levels deal with machine coordination and process control. ISAM integrates and distributes deliberative planning and reactive control functions throughout the entire hierarchical architecture, at all levels, with all spatial and temporal scales.

Introduction

The Intelligent Systems Architecture for Manufacturing (ISAM) is a reference model architecture for intelligent manufacturing systems. It is intended to provide a theoretical framework for the development of standards and performance measures for intelligent manufacturing systems. It is also intended to provide engineering guidelines for a wide variety of manufacturing applications.

The ISAM model addresses the manufacturing enterprise at a number of levels of abstraction.

1. At the highest level of abstraction, ISAM provides a conceptual framework for viewing the entire manufacturing enterprise as an intelligent system consisting of machines, processes, tools, facilities, computers, software, and human beings operating, over time, on materials to produce products.

2. At a lower level of abstraction, ISAM provides a reference model architecture to support the development of standards and performance measures, and the design manufacturing systems and software.

3. At a still lower level of abstraction, ISAM is intended to provide engineering guidelines to implement specific instances of manufacturing systems such as machining and inspection systems.

I. ISAM as a conceptual framework

The ISAM conceptual framework spans the entire range of manufacturing operations, from those that take place over time periods of microseconds and distances of microns to those that take place over time periods of years and distances of many kilometers. The ISAM model is intended to allow for the representation of activities that range from detailed dynamic analysis of a single actuator in a single machine to the combined activity of thousands of machines and human beings in hundreds of plants comprising the operations of a multinational corporation.

To span this wide range of activities, ISAM adopts a hierarchical layering with different range and resolution in time and space at each level. This permits the definition of functional entities at each level within the enterprise such that each entity can view its particular responsibilities and priorities at a level of spatial and temporal resolution that is understandable and manageable to itself. At any level within the hierarchy, functional entities receive goals and priorities from above and observe situations in the environment below. In each functional entity at each level, there are decisions to be made, plans to be formulated, and actions to be taken that affect peers and subordinates at levels below. Information must be processed, situations analyzed, and status reported to peers and supervisors above. Each functional entity needs access to a model of the world that enables intelligent decision making, planning, analysis, and reporting activity to be carried out despite the uncertainties and unwanted signals that exist in the real world. At each level, there are values (often not explicitly stated) that set priorities and guide decision making.

Typically, a manufacturing enterprise is organized into management units such that each management unit consists of a group of intelligent agents (humans or machines). Each of these possesses a particular combination of knowledge, skills, and abilities. Each has a job description that defines duties and responsibilities. Each management unit accepts tasks from higher level management units and issues sub tasks to subordinate management units. Within each management unit, agents are given job assignments and allocated resources with which to carry out their assignments. Within each management unit, intelligent agents schedule their activities to achieve the goals of the jobs assigned to them. Each agent is expected to make local executive decisions to keep things on schedule by solving problems and compensating for minor unexpected events.

Typically, each unit of management has a model of the world environment in which it must function. This world model is a representation of the state of the environment and of the entities that exist in the environment, including their attributes and relationships, and the events that take place in the environment. The world model also typically includes a set of rules that describes how the environment will behave under various conditions. Each unit of management also has access to sources of information that keep its world model current and accurate. Finally, each management unit has a set of values, or cost functions, that it uses to evaluate that state of the world and by which its performance is evaluated.

ISAM vs. Current Practice

For the most part, current industry practice assumes that manufacturing consists of largely predictable processes that can safely proceed without the benefit of, or need for, on-line measurement and real-time feedback control. Most adjustments in manufacturing processes are made by operators that often use intuition and experience to tune parameters. In control parlance, most production processes operate "open loop.” There is little or no consideration given to the need for real-time planning or replanning, automatic error recovery, on-line optimization, or adaptability to changing conditions. On-line schedule changes, and process modifications are handled mostly by manual ad-hoc methods.

Current interface standards are mostly limited to data exchange standards for static data such as IGES and STEP [1,2]. Most communication protocol standards such as MMS and TCP/IP [3,4] do not deal with the semantics or pragmatics of the processes being controlled. A few steps toward standards for dynamic interfaces which convey meaning are appearing, such as APIs and CORBA [5,6]. However, open architecture interface standards, based on standard functional decomposition, that can enable software from a variety of vendors to work together dynamically, are still well in the future.

Future manufacturing will be characterized by the need to adapt to the demands of agile manufacturing, including rapid response to changing customer requirements, concurrent design and engineering, lower cost of small volume production, out-sourcing of supply, distributed manufacturing, just in-time delivery, real-time planning and scheduling, increased demands for precision and quality, reduced tolerance for error, in-process measurement and feedback control. These demands generate requirements for adaptability and on-line decision making that cannot be met with current practice. Future manufacturing systems will require intelligent control concepts that for the most part are being developed outside of the field of industrial engineering. The ISAM conceptual framework attempts to apply intelligent control concepts [7,8,9] to the domain of manufacturing so as to enable the full range of agile manufacturing concepts.

II. ISAM as a reference model architecture

ISAM defines the functional elements, subsystems, interfaces, entities, relationships, and information units involved in intelligent manufacturing systems.

Df: **functional elements**

the fundamental computational units of a system

Axiom 1: The functional elements of an intelligent system are behavior generation, sensory perception, world modeling, and value judgment.

Axiom 2: World modeling maintains and uses a distributed dynamic store of knowledge that collectively forms a world model, that includes both a model of the manufacturing environment and a model of the system itself.

Axiom 3: The functional elements and knowledge database of an intelligent system can be represented in an architectural node by a set of modules interconnected by a by a communication system that transfers information between them.

Axiom 4. The complexity inherent in intelligent systems can be managed through hierarchical layering.

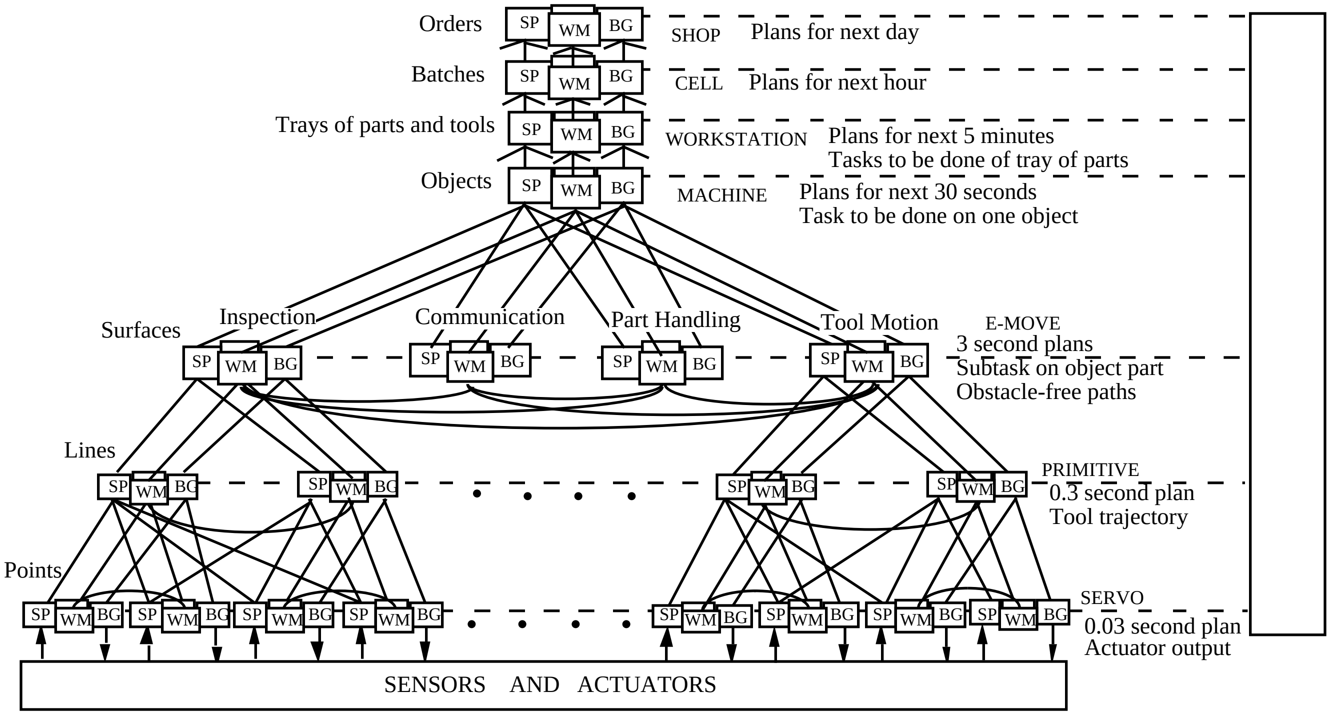

Figure 1. An ISAM reference model architecture for a machining center. Processing nodes are organized such that the BG modules form a command tree. Information in the KD is shared between WM modules in nodes within the same subtree. KD modules are not shown in this figure. The right side shows examples of the functional characteristics of the BG modules at each level. The left side shows examples of the type of entities recognized by the SP modules and stored by the WM in the KD knowledge database at each level. Sensory data paths flowing up the hierarchy typically form a graph, not a tree. VJ modules are hidden behind WM modules. An operator interface provides input to, and output from, modules in every node.

An example of an ISAM reference model architecture for an individual intelligent machine (a flexible machining center) is shown in Figure 1. This diagram consists of a hierarchy of control nodes, each of which contain modules corresponding to the functional elements defined in Axiom 1. That is, each node consists of a behavior generation (BG), world modeling (WM), sensory perception (SP), and knowledge database (KD) module (not shown in Figure 1.) Most nodes also contain a value judgment (VJ) module (hidden behind the WM module in Figure 2.) Each of the nodes is therefore an intelligent controller. An operator interface may access modules in all nodes at all levels.

Figure 1 illustrates a machining center with four subsystems: tool motion, part handling, communication, and inspection. Each of the four subsystems has one or more mechanisms. Each of these has one or more actuators and sensors. For example, the manipulation subsystem may consist of a spindle and tool path controller with several axes of continuous-motion path control, plus a tool changer consisting of a tool carousel and a manipulator. Each of these has two or more actuators and sensors. The part handling subsystem might be a pallet feeding mechanism, which consists of a conveyor and buffer, pallet shuttle, and a pair of turn tables. The communication subsystem might consist of a message encoding subsystem, a protocol syntax generator, and communications bus interface. The inspection subsystem might consist of mechanisms that use cameras and touch probes to detect and track objects, surfaces, edges and points, and compute trajectories for probe drive motors, and pan, tilt, and focus actuators. All of these functions need to be coordinated in order to successfully achieve behavioral goals.

The operator interface provides the ability for the operator to start or stop the system at any time, to single step, to override the feed rate of the motion controller, to actuate or monitor any of the tool change or pallet feeding mechanisms. The operator interface can send or display information from the communications subsystem, or display any of the state variables in the world model in real-time. Using the operator interface, a human operator is able to run diagnostic programs in the case of failures, display control programs (plans) while they are being executed, generate graphics images of tool paths, display volumes to be removed from parts by numerical control

(NC) programs, and portray shaded images or wire frame models of parts with dimensions and tolerances indicated with overlays.

In Figure 1 three levels of control are shown above the node representing the individual machine controller. This represents the chain of command that exists in the manufacturing environment above the individual machining center. There, of course, may be more than three levels above the machine. In Figure 1, the nodes at the upper levels may have the same degree of branching that exists at the lower levels. But, this is not shown in order to simplify the diagram.

The horizontal curved lines between WM modules represent the sharing of state information between nodes within sub trees in order to synchronize related tasks.

The functionality of each level in the ISAM reference model hierarchy is defined by the characteristic timing, bandwidth, and algorithms chosen for decomposing tasks and goals at each level. Typically these are design choices that depend on the dynamics of the processes being controlled. The numbers shown in Figure 1 are representative of those appropriate for a machining center. For other types of systems, different numbers would be derived from design parameters.

An ISAM Example

To illustrate the types of issues that can be addressed by the ISAM reference model architecture, an example of a seven level ISAM hierarchy for a machine shop is given below.

Level 7 -- Shop

The shop level receives input task commands of the form <manufacture_orders>. It decomposes those commands into output subtask commands to the cell level of the form <machine_orders>, <assemble_orders>, <ship_orders>, and <maintain_inventory>.

The shop level plans activities and allocates resources for one or more manufacturing cells for a period on the order of one eight hour shift. At the shop level, orders are sorted into batches, and a production schedule is generated for the cells to process the batches. At the shop level, the world model maintains a knowledge database containing names, contents, and attributes of batches and the inventory of tools and materials required to manufacture them. Maps may describe the location of, and routing between, manufacturing cells. Sensory perception processes compute information about the flow of parts, the level of inventory, and the operational status of all the cells in the shop. Value judgment computes the cost and benefit of various batching and routing options and calculates statistical quality control data. An operator interface allows human operators to visualize the status of orders and inventory, the flow of work, and the overall situation within the entire shop facility. Operators can intervene to change priorities and redirect the flow of materials and tools. Executors keep track of how well plans are being followed, and modify parameters as necessary to keep on plan. The output from the shop level provides work flow assignments for the cells.

Level 6—Cell

The cell level receives input task commands of the form <machine_orders>, and <assemble_orders>. It decomposes those commands into output subtask commands to the workstation level of the form <mill_batch>, <drill_batch>, <grind_batch>, <weld_batch>, <assemble_batch>, <inspect_batch>, and <transport_batch>.

The cell level plans activities and allocates resources for one or more workstations for a period of about an hour into the future. Batches of parts and tools are scheduled into particular workstations. The world model symbolic database contains names and attributes of batches of parts and the tools and materials necessary to manufacture them. Maps describe the location of, and routing between, workstations. Sensory perception determines the location and status of trays of parts and tools. Value judgment evaluates routing options for moving batches of parts and tools. An operator interface allows human operators to visualize the status of batches and the flow of work through and within the cell. Operators can intervene to change priorities and reorder the plan of operations. Executors keep track of how well plans are being followed, and modify parameters as necessary to keep on plan. The output from the cell level are commands issued to particular workstations to perform machining, inspection, or material handling operations on particular batches or trays of parts.

Level 5—Workstation

The workstation level receives input task commands of the form <mill-batch>, <drill_batch>, <grind_batch>, <weld_batch>, <assemble_batch>, <inspect_batch>, and <transport_batch>. It decomposes those commands into output subtask commands to the workstation level of the form <load_part>, <fixture_part>, <mill_part>, and <unload_part>.

The workstation level schedules tasks and controls the activities within each workstation with about a five minute planning horizon. A workstation may consist of a group of machines, such as one or more closely coupled machine tools, robots, inspection machines, materials transport devices, and part and tool buffers. Plans are developed and commands are issued to equipment to operate on material, tools, and fixtures in order to produce parts. The world model symbolic database contains names and attributes of parts, tools, and buffer trays in the workstation. Maps describe the location of parts, tools, and buffer trays. Sensory perception determines the position of parts and tools in trays and buffers. Value judgment evaluates plans for sequencing machining and parts handling operations within the workstation. An operator interface allows human operators to visualize the status of parts and tools within the workstation, or to intervene to change priorities and reorder the sequence of operations within the workstation. Executors keep track of how well plans are being followed and modify parameters as necessary to keep on plan. Output commands are issued to particular machine tools, robots, and tray buffers to perform tasks on individual parts, tools, and fixtures.

Level 4—Equipment task

The equipment level receives input task commands of the form <load_part>, <fixture_part>, <mill_part>, and <unload_part>. It decomposes those commands into output subtask commands to the E-move level of the form <locate_part>, <grasp_part>, <maneuver_part>, <set_clamps>, <mill_face>, <mill_pocket>, <mill_slot>, and <mill_champfer>.

The equipment level schedules tasks and controls the activities of each machine within a workstation with about a 30 second planning horizon. (Tasks that take much longer may be broken into several 30 second segments at the workstation level.) Level 4 decomposes each equipment task into elemental moves for the subsystems. Plans are developed that sequence elemental movements of tools and grippers, tool changers, and pallet shuttle systems. Commands are formulated to move tools and grippers so as to approach, grasp, move, fixture, cut, drill, mill, or measure parts. The world model symbolic database contains names and attributes of parts, such as their size and shape (dimensions and tolerances) and material characteristics (mass, color, and hardness.) Maps consist of drawings that illustrate part shape and the relative positions of part features. Sensory perception measures part dimensions and tolerances. Value judgment evaluates part quality and supports planning for part handling and fixturing sequences. An operator interface allows human operators to visualize the status of operations of the machine, or to intervene to change priorities or interrupt the sequence of operations. Executors keep track of how well plans are being followed and modify parameters as necessary to keep on plan. Output command are issued to level 3 for machining, manipulating, and inspecting part features.

Level 3—Elemental move (E-move)

The E-move level receives input task commands of the form <grasp_part>, <maneuver_part>, <set_clamps>, <mill_pocket>, and <mill_slot>. It decomposes those commands into output subtask commands to the primitive level of the form <move_gripper_along_path>, <move_part_along_path>, and <move_tool_along_path>.

The E-move level schedules and controls simple machine motions requiring a few seconds. (Motions that require significantly more time may be broken up at the task level into several elemental moves.) Plans are developed and commands are issued that define safe path way points for tools, manipulators, and inspection probes so as to avoid collisions and singularities, and assure part quality and process safety. The world model symbolic database contains names and attributes of part features such as surfaces, holes, pockets, grooves, threads, chamfers, and burrs. Maps consist of drawings that illustrate feature shape and the relative positions of feature boundaries. Sensory perception measures dimensions of individual features, and computes surface properties. Value judgment supports planning of machine motions and evaluates feature quality. An operator interface allows a human operator to visualize the state of the machine or to intervene to change mode or interrupt the sequence of operations. Executors keep track of how well plans are being followed, and modify parameters as necessary to keep on plan. Output consists of commands to move along trajectory segments between way points.

Level 2—Primitive

The primitive level receives input task commands of the form <move_{part | tool | gripper}_along_path>. It decomposes those commands into output subtask commands to the servo level of the form <go_to_(x, y, z, roll, pitch, yaw) at feedrate_v>.

The primitive level plans paths for tools, manipulators, and inspection probes so as to minimize time and optimize performance. It computes tool or gripper acceleration and deceleration profiles taking into consideration dynamical interaction between mass, stiffness, force, and time. Planning horizons are on the order of a few hundred milliseconds. The world model symbolic database contains names and attributes of linear features such as lines, trajectory segments, and vertices. Maps (when they exist) consist of perspective projections of linear features such as edges, lines, or tool or end-effector trajectories. Sensory perception computes observed motions of tools and grippers. Value judgment supports trajectory optimization. An operator interface allows a human operator to visualize the state of the machine, or to intervene to change mode or override the feed rate. Executors keep track of how well plans are being followed and modify parameters as necessary to keep within tolerance. Output consists of commands to move tool or grippers at desired velocities and accelerations, or to exert desired forces, or maintain desired stiffness parameters.

Level 1—Servo level

The servo level receives input task commands of the form <go_to_(x, y, z, roll, pitch, yaw) at feedrate_v>. It decomposes those commands into output subtask commands to the actuators in the form of desired force, torque, or power.

The servo level transforms commands from tool path to joint actuator coordinates. Planners interpolate between primitive trajectory points for each actuator with a planning horizon of a few tens of milliseconds. The world model symbolic database contains values of state variables such as joint positions, velocities, and forces, proximity sensor readings, position of discrete switches, state of touch probes, as well as image attributes associated with camera pixels. Maps consist of camera images and displays of sensor readings. Sensory perception scales and filters data from sensors that measure actuator positions, velocities, forces, torques, and touch. An operator interface allows a human operator to visualize the state of the machine, or to intervene to change mode, set switches, or jog individual axes. Executors servo individual actuators and motors to follow interpolated trajectories. Position, velocity, or force servoing may be implemented, and in various combinations. Output commands to power amplifiers specify desired actuator torque or power. Outputs are typically produced every few milliseconds (or whatever rate is dictated by the machine dynamics and servo performance requirements.) The servo level also commands switch closures that control discrete actuators such as relays and solenoids.

At the Servo and Primitive levels, the output command rate is typically clock driven on a regular cycle. At the E-Move level and above, the command output rate becomes irregular because it is event driven.

At each hierarchical level, World Modeling, Sensory Perception, and Value Judgment modules provide to the Behavior Generation modules the information needed for decision making and control. At each hierarchical level, Behavior Generation modules decompose tasks into sub tasks for subordinate Behavior Generation modules. At each level World Model knowledge is shared between Knowledge Databases at the same level, and relational pointers are established between Knowledge Data structures at both higher and lower levels.

At each level, commands and status information are transmitted up and down the control hierarchy and between Behavior Generation and Operator Interface modules. Operator Interfaces are able to display information from any of the functional modules and the Knowledge Database.

At each level, Sensory Perception modules accept sensory observations from Sensory Perception modules at lower levels, and output processed and clustered sensory observations to Sensory Perception modules at higher level nodes.

At all levels, SP modules compare and correlate predictions from the WM with observations from lower level SP modules. Differences are used to update the estimated state of the World stored in the world model Knowledge Database. Correlations are used to recognize correspondence between what is stored in the internal world model Knowledge Database and what is observed in the external real World.

At each level, knowledge is represented in a form and with a spatial and temporal resolution that meets the processing requirements of the node. At each level, there is a characteristic loop bandwidth, a characteristic planning horizon, a characteristic type of task decomposition, a characteristic range of temporal integration of sensory data, and a characteristic window of spatial integration. At each level, information is extracted from the sensory data stream to keep the world model knowledge database accurate and up to date.

At each level, sensory data is processed, entities are recognized, world model representations are maintained, and tasks are deliberatively decomposed into parallel and sequential sub tasks to be performed by cooperating sets of agents within the BG modules. At each level, feedback from sensors reactively closes a control loop allowing each agent to respond and react to unexpected events. At each level, tasks are decomposed into sub tasks and sub goals, and agent behavior is planned and controlled. The result is a system that combines and distributes deliberative and reactive control information throughout the entire hierarchical architecture with both planned and reactive capabilities tightly integrated at all levels of space and time resolution.

The specific numbers and functions given in this example are illustrative only. They are meant only to illustrate how the generic structure and function of the ISAM reference model architecture might be instantiated at the lower levels of a manufacturing enterprise. The point of this example is to illustrate how the ISAM multilevel hierarchical architecture integrates real-time planning and

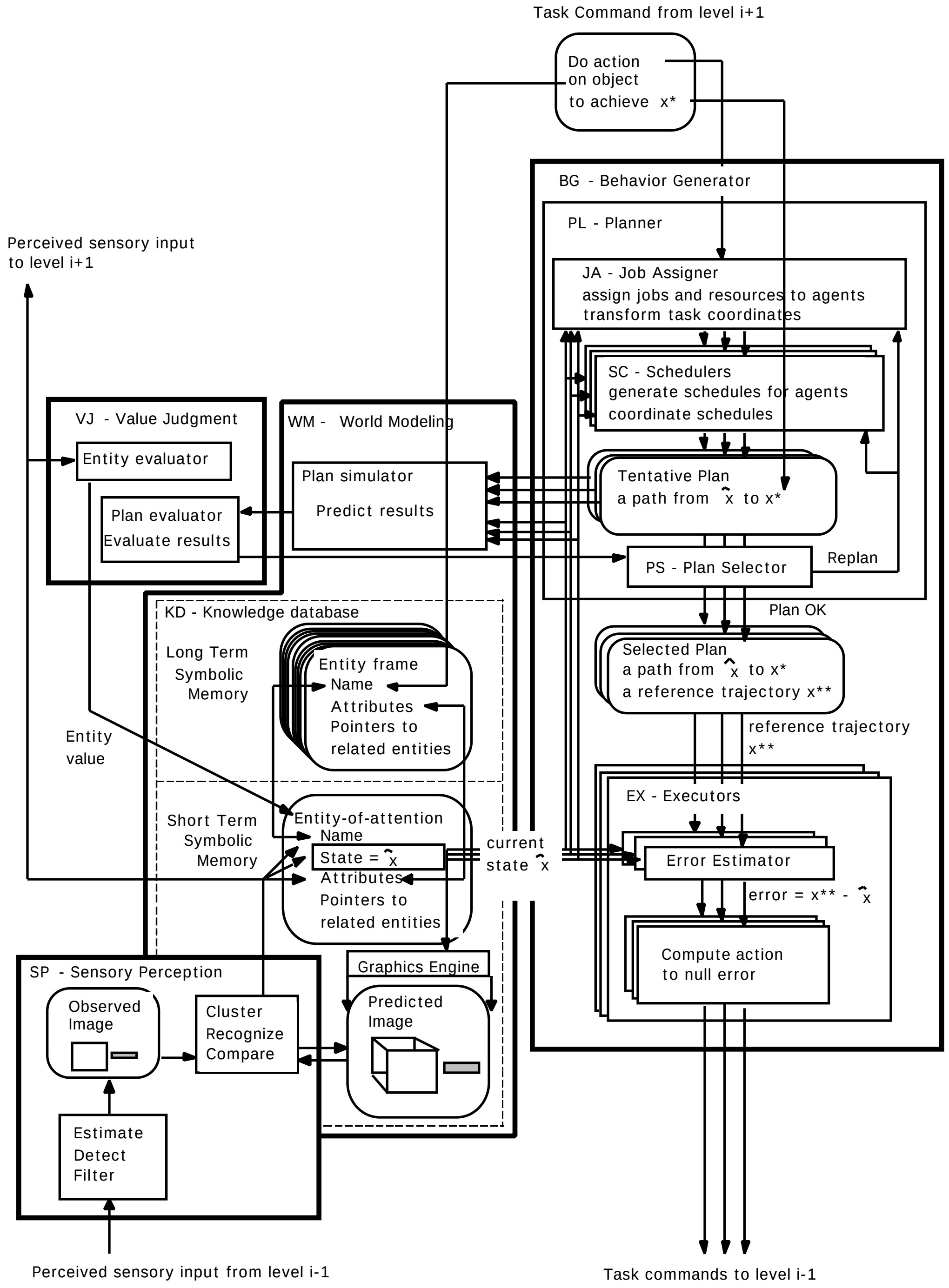

Figure 2. Relationships within a single node of the architecture A task command from level i+1 specifies the goal and the object. The object specification selects an entity from long term memory and moves it into short term memory. The BG module generates a plan that leads from the current state to the goal state. The EX sub modules execute the plan using estimated state information in the WM KD short term memory. The SP module processes sensory input from level i-1 to update the WM KD within the node, and sends processed information to level I+1.

execution of behavior with dynamic world modeling, knowledge representation, and sensory perception. At each level, behavior generation is guided by value judgments that optimize plans and evaluate results. The system architecture organizes the planning of behavior, the control of action, and the focusing of computational resources. The overall result is an intelligent real-time control system that is both driven by high level goals and reactive to sensory feedback.

A methodology for implementing the ISAM architecture for this and other types of specific manufacturing applications is currently being addressed at NIST as part of an ongoing project. As part of that effort, the internal structures of the BG, WM, SP, VJ, and KD modules are being developed.

III. ISAM as a implementation guide

A first step in designing an ISAM architecture for a large scale intelligent manufacturing system is to develop a generic processing node that can be duplicated many times at many different levels. The generic node can then be populated with the knowledge, skills, and abilities required by its particular duties and responsibilities, and interconnected with other generic nodes so as to produce the desired functionality.

The relationships and interactions between the BG, WM, KD, SP, and VJ modules in a generic node of the ISAM architecture are shown in Figure 2. The Behavior Generating (BG) modules contain sub modules Planner (PL) and Executor (EX). Planner has sub-submodules of Job Assignment (JA), Scheduling (SC) and Plan Selector (PS). The World Modeling (WM) module contains the Knowledge Database (KD), with both long-term and short-term symbolic representations and short term iconic images. In addition, WM contains a Simulator where the alternatives generated by JA and SC are tested for comparison in PS. The Sensory Processing (SP) module contains filtering, detecting, and estimating algorithms, plus mechanisms for comparing predictions generated by the WM module with observations from sensors. It has algorithms for recognizing entities and clustering entities into higher level entities. The Value Judgment (VJ) module evaluates plans and computes confidence factors based on the variance between observed and predicted sensory input.

Each node of the ISAM hierarchy closes a control loop. Input from sensors is processed through sensory processing (SP) modules and used by the world modeling (WM) modules to update the knowledge database (KD). This provides a current best estimate of the state of the world x. Depending on the concept of estimation used, a particular control law is applied to this feedback signal arriving at the EX sub module. The EX sub module computes the compensation required to minimize the eventual difference between the planned reference trajectory and the trajectory which emerges as a result of the control process. The value of "best estimate" is also used by the JA and SC sub-submodules and by the WM plan simulator to perform their respective planning computations. It is also used in generating a short term iconic image that forms the basis for comparison, recognition, and recursive estimation in the image domain in the sensory processing SP module.

Current work at NIST is directed toward the development of the generic node shown in Figure 2. A control hierarchy for a machine shop, including inspection machines, machine tools, and a material handling system is being constructed from such nodes interconnected through Application Program Interfaces (APIs) that are intended to serve as a prototype for future standards [10]. An Enhanced Machine Controller has been developed using APIs to serve as a testbed for open architecture controllers for machine tools, coordinate measuring machines, and robots [11]. Future efforts will be directed toward the development of engineering methods for implementation of intelligent manufacturing systems.

References

[1] Smith, B., Wellington, J., IGES - Initial Graphics Exchange Specification, Version 3.0, NBSIR 86-3359, April, 1986
[2] STEP - Standard for Exchange of Product Model Data
[3] MMS - Manufacturing Messaging Specification
[4] TCP/IP - Transmission Control Protocol / Internet Protocol
[5] API - Application Programming Interfaces
[6] CORBA - Common Object Request Broker Architecture http://www.acl.lanl.gov/CORBA/
[7] Albus, J.S., Meystel, A.M., "A Reference Architecture for Design and Implementation of Large Complex Systems", International Journal of Intelligent Control and Systems, Vol. 1, No. 1, 1996 p. 15-30
[8] Wallace, S., Senehi, M.K., Barkmeyer, E., Luce, M., Ray, S., Wallace, E., "Control Entity Interface Specification", NISTIR 5272, 1993
[9] Domnez, M.A., (Ed.), "Progress Report of the Quality in Automation Project for FY90, NISTIR 4536, 1991
[10] Technologies Enabling Agile Manufacturing (TEAM) API Standards http://isd.cme.gov/info/team
[11] Proctor, F.M., Michaloski, J., "Enhanced Machine Controller Architecture Overview," NISTIR 5331, 1993
