N88-17265

IMPLEMENTATION OF A ROBOTIC FLEXIBLE ASSEMBLY SYSTEM

Ronald C. Benton
Honeywell Corporate Systems Development Division
1000 Boone Avenue North
Golden Valley, MN 55427

ABSTRACT

As part of the Intelligent Task Automation (ITA) program sponsored by the Air Force and the Defense Advanced Research Projects Agency (DARPA), a team of industry, research institute and university members developed enabling technologies for programmable, sensory-controlled manipulation in unstructured environments. These technologies included 2-D/3-D vision sensing and understanding, force sensing and high-speed force control, 2.5-D vision alignment and control, and multiple-processor architectures. This paper describes the subsequent design of a flexible, programmable, sensor-controlled robotic assembly system for small electromechanical devices using these technologies and ongoing implementation and integration efforts. Using vision, the system will acquire parts dumped randomly in a tray. Using vision and force control, it will perform high-speed part mating, in-process monitoring/verification of expected results and autonomous recovery from some errors. It will be programmed off-line with semiautomatic action planning.

INTRODUCTION

The Intelligent Task Automation program objectives are to develop and demonstrate generic technologies that (1) form a basis for advanced intelligent automation systems, (2) have near-term application to unit processes, component assembly, and many aspects of defense batch manufacturing, and (3) establish the technology base for new opportunities to apply intelligent systems to complex military tasks [1, 2, 3]. This program identifies and develops robot-related technologies that should have a high payoff for future manufacturing systems.

Phase I included research, development and feasibility demonstrations that culminated in a conceptual design for each component as well as the system itself. Breadboard hardware and developmental software were constructed to critically address areas of uncertain technical risk and to show feasibility of the entire system. Phase I concluded in December 1985 with the results documented in References 4, 5 and 6.

Phase II of the program began 1 September 1986 and will culminate in mid-1988 with demonstration of an entire system. The functional elements of the Intelligent Task Automation System (ITAS) are shown in Figure 1 along with the enabling technologies developed during Phase I, providing the basis for system design. The ITAS demonstration configuration is shown in Figure 2. The ITAS is a flexible, programmable, sensor-controlled robotic assembly system for small electromechanical parts. It operates in an unstructured environment relative to today's technology since it recognizes, locates and grasps the proper parts (which may be touching or overlapping) to be assembled from a tray. Once parts are grasped, the ITAS performs high-speed mating and fastening actions to assemble the device. During assembly, it performs in-process monitoring and verification of expected results and autonomous recovery from some anticipated errors. The ITAS is programmed off-line with semiautomatic action planning enhancements.

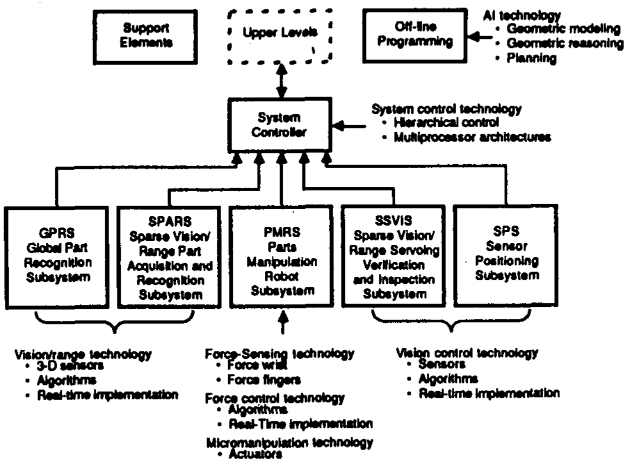

Support Elements
Upper Levels
Off-line Programming
AI technology
• Geometric modeling
• Geometric reasoning
• Planning
System Controller
System control technology
• Hierarchical control
• Multiprocessor architectures
GPRS Global Part Recognition Subsystem
SPARS Sparse Vision/ Range Part Acquisition and Recognition Subsystem
PMRS Parts Manipulation Robot Subsystem
SSVIS Sparse Vision/ Range Servoing Verification and Inspection Subsystem
SPS Sensor Positioning Subsystem
Vision/range technology
• 3-D sensors
• Algorithms
• Real-time implementation
Force-Sensing technology
• Force wrist
• Force fingers
Force control technology
• Algorithms
• Real-Time implementation
Micromanipulation technology
• Actuators
Vision control technology
• Sensors
• Algorithms
• Real-time implementation

Figure 1. ITAS Functional Elements and Associated Enabling Technologies

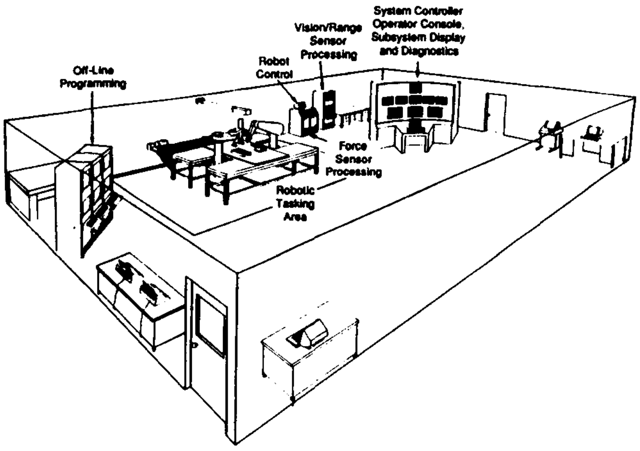

Off-Line Programming
Robot Control
Vision/Range Sensor Processing
System Controller Operator Console, Subsystem Display and Diagnostics
Force Sensor Processing
Robotic Tasking Area

Figure 2. ITA System

To demonstrate the primary assembly task, automatic assembly of a Honeywell 1EN1-6 microswitch will be performed (see Figure 3). This assembly was selected due to the variety of part geometries, materials and mating requirements. A different electromechanical device will also be assembled to illustrate the flexibility of the ITAS.

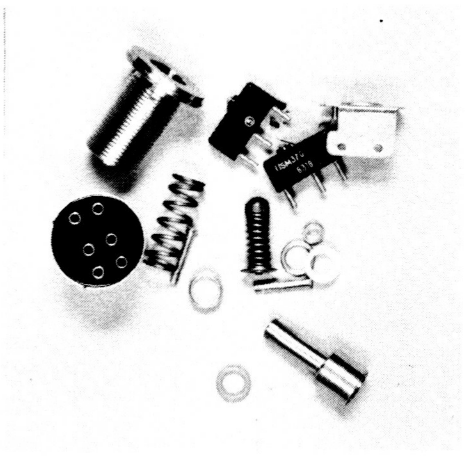


Figure 3. Tray of 1EN1 Microswitch Parts as Seen by Global Part Recognition Sensor

DEVELOPMENT APPROACH

The ITA Phase II System development cycle follows the classic systems engineering approach of requirements formulation, design, implementation, integration and test. Since most of the effort involves software, the structured analysis/structured design method was selected [7]. The graphics-based SA/SD process consists of developing both a system data/control flow model (by analyzing the requirements of the system) and a system structure model that is a translation of those requirements into a description of the system architecture, system components, interfaces, data and materials necessary for the implementation phase, i.e., a design. The results are documented in so-called "structured" specifications.

A computer-aided software engineering tool set from Interactive Development Environments Inc. (IDE) [8] running on UNIX-based Sun Microsystems work stations was applied to support the SA/SD methodology. Using graphics input, the tool generates a system data base to enable automatic decomposition, consistency checking, dictionary generation, and on-line specification access. To permit specification generation, the IDE tool was linked to a documentation system to automatically extract textual descriptions attached to graphic diagrams and to insert that text in proper format in the specification documents.

This "automatic" documentation tool set was developed to simplify creation of documents for the ITA System; i.e., specifications, manuals, etc. The documents can be large, can be constructed from many parts in complex ways (e.g., graphics and text from the SA/SD tools must be integrated into design specifications), must be controlled (configuration management), and must follow strict style guidelines. Since the SA/SD tools reside on UNIX-based Sun Microsystems work stations and most of the code will be generated there, the documentation tools were built using UNIX tools. Document content and style guidelines are supported by on-line templates for each document type, a set of *nroff/troff* formatting macros and some preprocessing of the documents. Since the templates are ASCII files, both designers and secretaries fill them in using terminals and text editors of their choice, and the documentation system generates a document in the proper format. Finally, to aid in control of the document as a whole, procedures for revision control were developed using the UNIX *SCCS* (Source Code Control System).

Ideally, it would be possible to perform a complete top-down, implementation-independent requirements analysis before considering design candidates. However, in many cases, as in ITA, preliminary and/or partially documented implementations exist that could, at least in part, be used in the new design. Using the structured design tools described above, reverse engineering of these implementations is being performed in parallel to support design activities. Specifically, the objectives of this process are to: (1) generate top-level maps of major modules and calling-tree structures; (2) identify "black box" modules, i.e., code with sufficient maturity and documentation for reuse as is; (3) map calling structures of high risk and/or critical code; (4) identify potential trouble spots; (5) provide input to design decisions on candidate selection, reusability, and high-payoff rework areas.

The process is based on generation of a "battle map" using annotated structure charts. Each module is rated on the level of existing documentation, a subjective "modifiability" index (i.e., how easy it would be to modify the code, if necessary) and the extent of global variable usage. The battle map enables quick identification of both black box modules and trouble spots.

The CIMStation work cell simulation tool from Silma Inc. was installed on an Apollo work station to support system analysis, design, incremental integration and off-line programming throughout Phase II [9]. CIMStation provides a geometric modeler with IGES interface, dynamic robot simulations, multiple process simulation with synchronized signal/wait interprocess communication, and a LISP-based, Pascal-like, device-independent simulation language with device-dependent translators. A rapid prototype of the ITA System design was developed to validate the subsystem functional allocations, verify robot selections, improve the system time-line estimates and identify task-level programming approaches.

Finally, rapid prototyping of critical hardware/software elements, such as the force-controlled robot and vision sensors, is being performed where necessary to support the design process.

SYSTEM REQUIREMENTS AND DESIGN

The ITA System will implement generic technologies developed in vision/range sensing and understanding, force sensing and understanding, sensory-based control and manipulation, and off-line programming (see Figure 1). A primary assembly task (i.e., the 1EN1 microswitch) will be performed to demonstrate its capability, and secondary assembly tasks will be performed to demonstrate its flexibility. The autonomous performance of these tasks along with a demonstration of the ITAS reprogrammability must clearly distinguish between the ITA System's method of performing a task and that of hard automation.

The capability of the ITA System to become part of a larger system must also be demonstrated. In a factory of the future, the ITAS would be one of many work cells at the lowest levels of the hierarchy, such as in a small batch assembly plant. Here, an automated storage/retrieval work cell would generate the kits of parts for each device to be assembled. The ITAS would assemble these kits and send them to other work cells for subsequent assembly operations, final inspection and packaging (or, if necessary, rework). These cells would be interconnected with an intercell materials handling system such as conveyors and/or robot carts. Each group of work cells would be coordinated by the supervisor layer of the factory hierarchy to fill customer orders for small batches of devices.

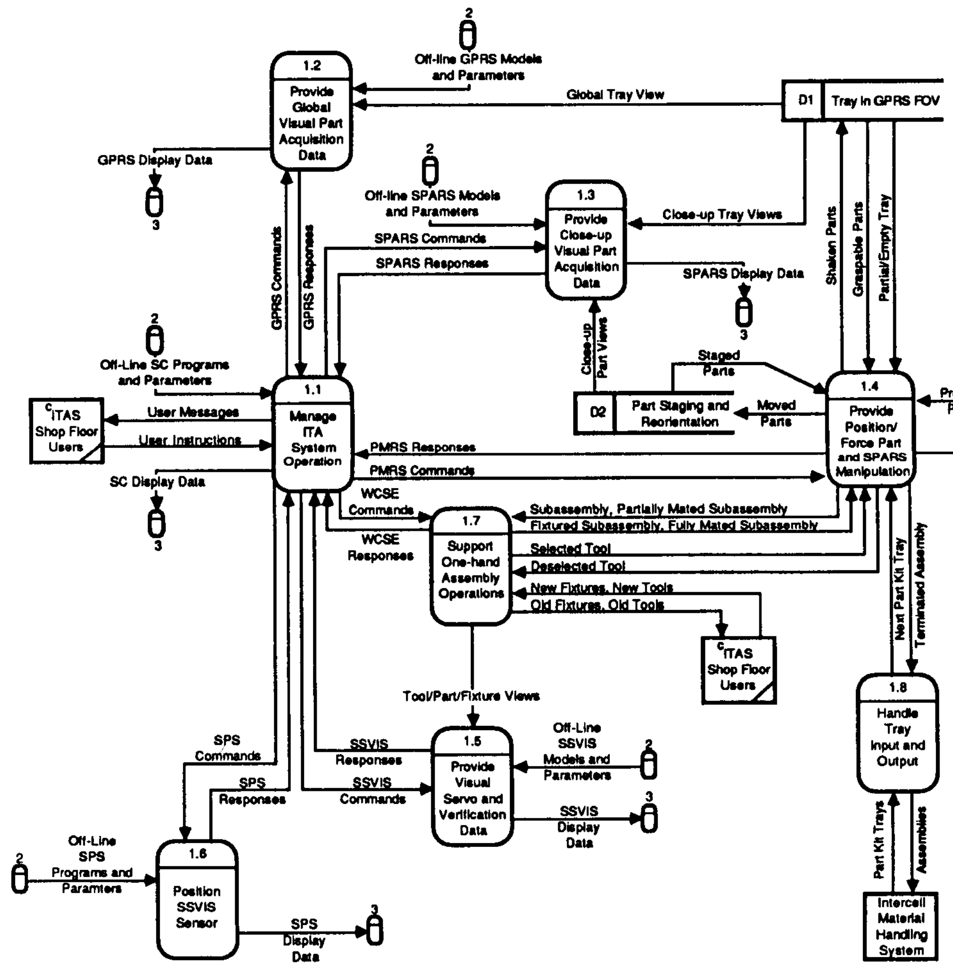
These cells would be tended by shop floor operators who would support changeover from one type of assembly to another and correct noncatastrophic equipment failures. Routine maintenance, catastrophic failures, and test/debug of first-time assembly tasks would be handled by manufacturing technicians/engineers. To maximize utilization of shop floor equipment, however, work cell programming would be performed off-line by CAD/CAM specialists with access to computer-based information about the new device to be manufactured and the capabilities of the shop floor equipment. Thus, work cell software and, if necessary, new tools/fixtures would be developed without taking shop floor equipment out of production.

Since the ITAS has three operating modes–on-line (i.e., the shop floor segment of the ITAS is producing assemblies), downtime (all shop floor activities when the ITAS is not on-line), and off-line programming (which occurs independently of the other two modes)–the external performance requirements for each mode were defined. In on-line mode, the system will perform assembly of the 1EN1-6 microswitch and other secondary assembly tasks in an autonomous manner with speed and reliability comparable to a human. It will: (1) recognize, locate and grasp the proper parts to be assembled, which are presented randomly in a tray and may be touching or overlapping; (2) perform high-speed mating and fastening actions to combine these parts into a subassembly using simple and/or multifunction fixtures; (3) provide in-process monitoring and verification of expected results during the assembly; (4) provide annunciation, recording and error recovery from expected (i.e., previously defined as potential) error conditions; (5) provide annunciation and recording of all unexpected results.

Downtime activities include ITAS start-up, shutdown, routine maintenance, changeover and test/debug of new tasks. Thus, the duration and/or frequency of this mode should be minimized. The off-line programming mode will occur concurrently with the other two modes to provide maximum productivity in the shop floor segment of the ITAS. Thus, both the amount of programming time and expertise required off-line and the time spent in the downtime mode for test/debug (to get the ITAS to perform a new assembly task) should be minimized.

The resulting system requirements flow model (context diagram, not shown) clarifies the ITAS boundaries. Part kit trays are provided by an intercell materials handling system (a human operator will be used in the Phase II demonstration); similarly, trays of completed or aborted assemblies are removed by this system. The two classes of users that interact with the ITAS are shop floor users (operators and production engineers) and off-line programmers. Decomposing one level reveals the two major segments of the ITAS–the shop floor, Process 1, and the off-line programming, Process 2. These processes can run concurrently and could be located in different sites with appropriate communication links. Process 3 generates displays to illustrate nonobvious aspects of ITAS operation for Phase II demonstration purposes. The decomposition of Process 1 is shown in Figure 4. The detailed operational scenario presented later illustrates how this flow model represents the ITA System's functional requirements.

The structure of the ITAS (see Figure 1) resulted from allocating one or more processes in the requirements flow model to "black



1.2 Provide Global Visual Part Acquisition Data
Off-line GPRS Models and Parameters
Global Tray View
D1 Tray in GPRS FOV
GPRS Display Data
Off-line SPARS Models and Parameters
SPARS Commands
SPARS Responses
1.3 Provide Close-up Visual Part Acquisition Data
Close-up Tray Views
SPARS Display Data
GPRS Commands
GPRS Responses
Close-up Part Views
Shaken Parts
Graspable Parts
Partial/Empty Tray
Off-Line SC Programs and Parameters
Staged Parts
D2 Part Staging and Reorientation
Moved Parts
Off-Line PMRS Programs and Parameters
ITAS Shop Floor Users
User Messages
User Instructions
1.1 Manage ITA System Operation
PMRS Responses
PMRS Commands
1.4 Provide Position/ Force Part and SPARS Manipulation
PMRS Display Data
SC Display Data
WCSE Commands
WCSE Responses
1.7 Support One-hand Assembly Operations
Subassembly, Partially Mated Subassembly
Fixtured Subassembly, Fully Mated Subassembly
Selected Tool
Deselected Tool
New Fixtures, New Tools
Old Fixtures, Old Tools
ITAS Shop Floor Users
Next Part Kit Tray
Terminated Assembly
Tool/Part/Fixture Views
1.8 Handle Tray Input and Output
SPS Commands
SSVIS Responses
1.5 Provide Visual Servo and Verification Data
Off-Line SSVIS Models and Parameters
SPS Responses
SSVIS Commands
SSVIS Display Data
Off-Line SPS Programs and Paramters
1.6 Position SSVIS Sensor
SPS Display Data
Part Kit Trays
Assemblies
Intercell Material Handling System

Figure 4. Provide Shop Floor Assembly Functions (Process 1.0)

ORIGINAL PAGE IS
OF POOR QUALITY

box" system elements with well-defined interfaces. These system elements, or subsystems, interact to achieve the overall system requirements as shown by comparison to Figure 4:

- A System Controller (SC)--The controller will provide the upper-level control to coordinate three vision subsystems and two robots (Process 1.1).
- A Global Part Recognition Subsystem (GPRS)--This subsystem will contain the primary part recognition sensors to acquire and process gray-scale and dense vision/range images for part identification, location and acquisition strategy generation (Process 1.2).
- An Arm-Mounted Sparse Vision/Range Part Acquisition and Recognition Subsystem (SPARS)--This subsystem will provide the functions to assist GPRS during part recognition and to fine tune part location prior to acquisition (Process 1.3).
- A Part Manipulation Robot Subsystem (PMRS)--This subsystem with force sensing and control will provide the delicate parts manipulation function and will perform part acquisition, mating, and fastening operations along with contact verification of in-process conditions before and after part-mating operations (Process 1.4).
- An Arm-Mounted Sparse Vision/Range Servoing Verification and Inspection Subsystem (SSVIS)--This subsystem will measure relative part misalignment during mating operations and conduct noncontact inspection/verification of in-process conditions before and after part-mating operations (Process 1.5).
- A Sensor Positioning Subsystem (SPS)--This subsystem will position the SSVIS sensor (Process 1.6).
- An Off-Line Programming Subsystem (OLPS)--This subsystem will perform interactive system programming, program validation, program download and data base generation for the on-line segments of the system (Process 2).
- Work Cell Support Elements (WCSE)--This subsystem will provide the work table, the parts presentation means, the fixtures used during assembly and the structure to support the sensors (Process 1.7 and Process 1.8)

The major hardware components of the subsystems described are configured as shown in Figure 5.

OPERATIONAL SCENARIO

The ITAS requirements and design may be better understood (and verified) by walking through the intended operational scenario for each of the three operating modes.

On-Line Mode

Prior to entering this mode, all system programs and data bases for the desired assembly task (for example, the 1EN1-6 microswitch) will have already been generated as described later. In addition, they will have been tested and debugged on the shop floor segment of the ITAS and loaded along with any tools and fixtures.

The operator then invokes a command such as "ASSEMBLE n 1EN1-6," and the SC (Process 1.1) will begin sequencing the system to assemble the n microswitches. The PMRS robot (Process 1.4) will be directed to acquire from the conveyor (Process 1.8) the 5-inch by 5-inch trays (into which kits of the 17 microswitch parts have been dumped) and to position each tray beneath the GPRS sensor, where PMRS uses force sensing to seat the tray against registration stops (Data Store "Tray in GPRS FOV"). No parts presentation means other than the tray will be used. The SC directs GPRS (Process 1.2) to find the desired part or any of an ordered list of larger parts. GPRS then acquires gray-scale and dense-range (terrain map) images to recognize, locate and determine a grasp strategy (i.e., a PMRS finger selection and corresponding unobstructed grasp points) to acquire the parts in the order needed for assembly. The parts will be touching, overlapping and possibly occluding each other in the tray; thus, if GPRS fails to find the desired part (e.g., the first part needed for the microswitch, the bushing, is large, but the second part, the red O-ring, is small and frequently occluded), the first larger part found is acquired and placed in the staging area (Data Store "Part Staging and Reorientation"), and GPRS is directed again to find the desired part. If GPRS fails to find any of the parts in the ordered list, the SC directs PMRS to shake the tray and reposition it under GPRS. If GPRS still fails, the operator is notified.

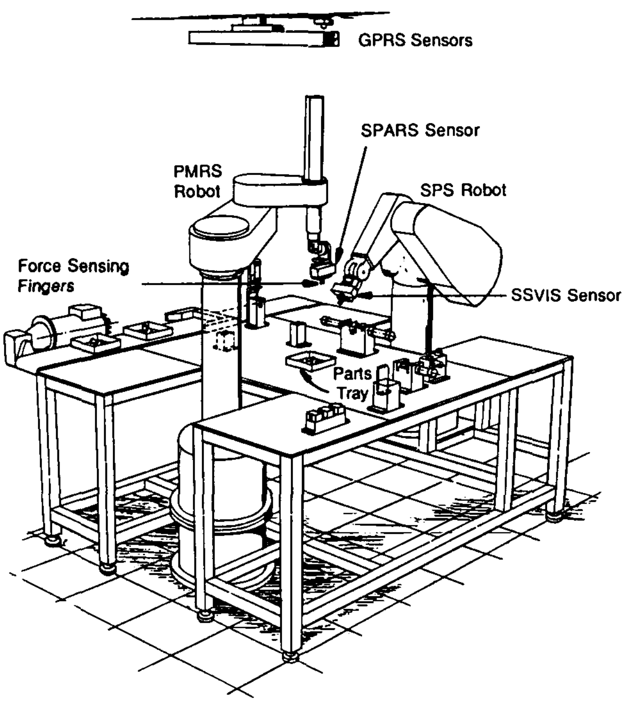


Figure 5. ITAS Robotic Tasking Area

Whenever PMRS is available for acquiring another part from the tray (i.e., not doing an assembly operation), the SC asks GPRS what it has found so far in its list of parts. GPRS replies but continues to work on its ordered list until every part is found, it gives up, or the SC sends GPRS a new list. Thus, if subsequent operations find that GPRS' reply is unacceptable, the SC can return to GPRS to see if a better answer has been found. For example, GPRS will tag some of the parts it has found for SPARS (Process 1.3) to verify the identity and/or pose of a given part at a given location in the tray. Then, while GPRS continues to work, SPARS will be positioned over the part by PMRS, where it will gather vision/range data and provide the appropriate feedback. If SPARS rejects GPRS' answer, the SC can go back to GPRS for another part(s) to acquire.

Thus, the PMRS robot will acquire the quick-change fingers (from Process 1.7) directed by GPRS to pick up the part, use its servoed gripper to open the fingers to an appropriate size, use a force-controlled guarded move to check for collisions while placing its

ORIGINAL PAGE IS
OF POOR QUALITY

fingers around the part, close the gripper with a grip force appropriate for the part, verify that its finger opening matches the part dimensions, and remove the part from the tray. If any of the above part identity/grasp strategy verifications fails, the SC will return to GPRS for another part(s). PMRS may be directed to place the part in the staging area to regrasp it properly or flip it over for the upcoming assembly steps; SPARS may be required to verify/fine tune the part pose to enable PMRS to regrasp it. Once PMRS has cleared the parts tray, GPRS will be sent a new list of parts.

Meanwhile, the SPS robot (Process 1.6) will be posing the SSVIS sensor (Process 1.5) at the appropriate fixture/subassembly (Process 1.7) for the upcoming part assembly operation. PMRS will blindly bring the two parts to be assembled into the appropriate relative poses necessary for mating. SSVIS will visually measure the relative misalignment between these parts and provide feedback to PMRS (via the SC) to correct the misalignment. When the parts are aligned, SSVIS will verify that the in-process conditions are correct for the assembly to proceed (i.e., correct parts and tools are present). If preassembly conditions are acceptable, PMRS will mate the two parts using high-speed force servoing for compliant, delicate micromanipulation of the parts. Mating operations will terminate when position and/or force limits are reached; data from the termination will be interpreted for contact verification of in-process conditions. PMRS will extract its fingers from the subassembly and return to the tray to acquire the next part that GPRS has found while this assembly was occurring.

With PMRS out of the way, the SPS robot can position the SSVIS sensor wherever necessary (e.g., looking down the bushing bore) to verify post-assembly conditions (i.e., correct parts are present and mated correctly). If expected errors exist, the SC provides error correction based upon scripts generated during previous testing.

Each additional part is assembled into a microswitch subassembly in the above fashion. If an in-process error is detected where the SC has a predetermined error recovery script, the system will attempt to correct it; otherwise, the SC will gracefully halt the system, notify the operator, and provide system and assembly status displays to enable quick recovery. When the assembly is completed, the microswitch subassembly is placed in the tray and the tray is returned to the conveyor. The next tray of parts is then acquired and assembled into a microswitch. This continues until all n microswitches have been assembled.

Errors produced in fixture placement and off-line programming are handled by the ITA System sensors; thus each assembly can be one of a kind. However, when multiple assemblies of one kind are to be produced, system throughput can be improved by updating or "touching up" the SC data base. This is defined as system autotuning. Thus, when the operator replies to the SC that the proper tools/fixtures are in place, the system can be instructed to perform autotuning. GPRS, PMRS and SPS are calibrated relative to permanent objects in the work cell during routine setup (e.g., the GPRS parts tray coordinate frame is calibrated with respect to the permanent tray registration stops). However, when special fixtures/tools are placed in the work cell, the programs generated off-line must be touched up. Here, SSVIS is first used to verify the SPS pose, accommodating errors within its 1-cubic-inch field of view; these results are fed back to the SC through queries to SPS about its new pose. Then the normal assembly steps are performed on the first tray of 1EN1-6 parts at slow speed with guarded PMRS moves. Each part acquired from the tray is staged and regrasped to ensure a nearly ideal pose in the gripper. SSVIS is then used to correct any misalignment, and the results are fed back to the SC via queries to PMRS about its new pose. After the part-mating operation, the force sensor data is analyzed to provide fine-tuned parameters for that force-controlled operation. Subsequent assemblies will now proceed at a faster rate since fewer sensory corrections are required. SSVIS is still used to accommodate part misalignment in the PMRS hand and to provide in-process verification.

Downtime Mode

This mode occurs whenever the shop floor segment of the ITAS is not in on-line mode or powered off. Downtime mode activities begin with the operator starting up the ITAS; i.e., all subsystems are brought up through a "cold-boot" operation to one of listening for commands from the SC. The SC then verifies the communication integrity and the state of health for each subsystem. If maintenance is required (e.g., sensor or robot calibration), it is performed at this point.

To begin assembling a device (or to change over from one type to another), the operator invokes a simple command such as "INIT CELL 1EN1-6" at the SC console. This causes the SC to load its data base for the 1EN1-6 assembly and to invoke GPRS, SPARS and SSVIS to load their data bases; PMRS and SPS are instructed to load the appropriate programs. The SC then prompts the operator to supply any special fixtures and/or tools required for the 1EN1-6 assembly and where to place them. When the operator notifies the SC that the proper tools/fixtures are in place, the cell is ready to begin assembly in on-line mode.

Programs and data for new, first-time assembly tasks (generated off-line) are tested and debugged in downtime mode. The subsystems' programs and data are modified to achieve acceptable performance using the SC console and/or the appropriate subsystem(s) console.

Off-Line Programming Mode

System programming within the system's assembly domain is accomplished by generating or modifying high-level task control programs and by generating data bases (Process 2). The three data bases for the vision subsystems are interactively generated off-line using training data collected from the on-line vision sensors and preprocessors. A work cell simulator is used to develop and test programs off-line for the robots and the SC and to generate the SC data base.

SC and Robot Programming--CAD models of the parts are generated with a coordinate system defined for each part. These parts are then "assembled" on the CAD system to compute their relative poses when properly assembled. Any tools and fixtures required for this assembly and not already in the ITA System work cell simulation are modeled. All new part, tool and fixture models are then passed to the work cell simulation.

The work cell simulation now contains geometric models of all components in the work cell: robots, fixtures and fingers/tools as well as the parts and the desired subassembly (see Figure 6). It also contains a functional simulation of the system with detailed simulations of the SC and the two robots and their controllers and simplified simulations of GPRS, SPARS and SSVIS. Since the robot controllers obtain all their task-specific data from the SC through procedure call parameters, only programs and data base entries for the SC must be written and tested for the desired assembly task. With the addition of user-specified safe approach/departure points and paths, all geometric information required to program the assembly task is available on the simulation. The resulting programs and data structures that are sufficient to run the simulation correctly are then used to generate the SC data base via a software language translation interface.

After the geometric modeling is complete, the user first defines safe approach/departure points and paths for the tray and all assembly fixtures interactively. Thus, poses like TRAY_SAFE and FIXTURE_1_SAFE become known to the simulation, along with

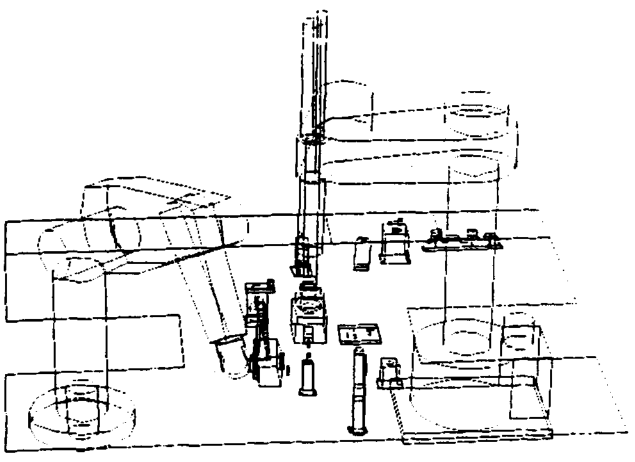


Figure 6. Simulation of ITA System Using CIMStation Tool

lists of "via points" for safe paths like TRAY_TO_FIXTURE_1. The parts are then placed interactively in various poses in the tray.

The user then enters the assembly operators such as LOCATE BUSHING, GRASP BUSHING FROM TRAY, PLACE BUSHING IN FIXTURE_1. As each operator is simulated, the system prompts the user if additional information is required. For instance, LOCATE BUSHING initially causes no prompts; the GPRS simulation simply queries the geometric modeler to provide the pose of the bushing--functionally equivalent to the real GPRS's recognition/location function. However, to determine a grasp strategy, the LOCATE operator lacks information: part-pose label, finger type, finger opening size and grasp position. Therefore, the user is prompted to supply the part-pose label (e.g., BUSHING_PRONE), which enables the simulated LOCATE operator to access its version of the GPRS grasp data base (created separately, as shown below) to obtain the additional data. GRASP BUSHING FROM TRAY then has all the data it needs except grip force, which is entered by the user. PLACE BUSHING IN FIXTURE_1 will prompt the user to "place" the bushing in Fixture 1 by interactively placing the PMRS robot, which is holding the bushing, in the desired pose. Operators like INSERT will require data to be entered such as SSVIS/SPS sensor pose, alignment tolerances, natural and artificial constraints for part mating, and force/position limits. Once the entire assembly sequence has been programmed, the parts are "placed" (using the simulation) in various poses in the tray to test the program and data. The verified assembly task program and data base are then translated and downloaded to the SC.

GPRS Data Base Generation--The GPRS data base is generated interactively off-line using the features extracted from training images of the parts in their various poses in the tray. These training images and features will have been gathered and stored using the GPRS on-line sensors and feature detection processing. First, an image of the part to be trained is selected and a coordinate system is defined for the part that corresponds to the one generated for the SC. Features detected for the part from both gray-scale and terrain map data are interactively selected to generate preliminary recognition models. Recognition strategies (e.g., which sensors to use, which part pose to look for first) are determined by the user. These preliminary models are then refined by exercising them on other training images of the same part in the same pose; during this process the system gathers statistics to adjust the model tolerances appropriately.

SPARS/SSVIS Data Base Generation--The SPARS and SSVIS data bases are generated in similar fashion to GPRS. To gather training data, a copy of the SPARS/SSVIS sensors on a six-degree-of-freedom repositionable fixture emulates the PMRS and SPS robots. First, data are gathered from the parts tray for training the SPARS subsystem. Then, a five-degree-of-freedom positioning fixture that can hold the quick-change fingers/tools to be used by PMRS gathers data for the SSVIS subsystem. Model building then proceeds in a fashion similar to that of GPRS.

SUBSYSTEM DETAILED DESIGN

The detailed design for all of the subsystems described above is currently nearing completion, and implementation of "certified" subsystems and/or their components has begun. Highlights of the designs are presented below.

System Controller

The primary function of the SC is to control, coordinate and monitor the subsystems that make up the ITA System. To maximize the system's effectiveness at its tasks, the SC must also manage the concurrent use of system resources; detect, respond to and correct error conditions whenever possible; and notify the shop floor user in the event of an unrecoverable error. All system, configuration and assembly data will be maintained in a "world model" of the ITAS within the SC. Facilities for development of new assembly applications, for editing existing assembly programs, and for system maintenance and calibration are also part of the SC functionality. The SC will provide test and debugging facilities for use in integrating subsystems into the ITA System during system development.

User Interface--The user interface requirements for each of the SC operating modes were determined using Reference 10 (among others) as a design guideline for sequence control, user guidance, data display, data entry and data protection. Since menus/forms are the most straightforward candidates for implementing the interface, requirements based on these candidates can be readily applied to others. Sample menus were generated to define requirements.

Error Recovery--Three levels of sophistication were envisioned. To recover after detecting an error condition, the SC could notify the operator, execute preprogrammed scripts of assembly operators/primitives, or execute commands from an "expert advisor." Three error scenarios were analyzed to determine the requirements of automatic recovery: (1) GPRS cannot find the desired part in the tray; (2) during preassembly verification, SSVIS indicates that the part is not in PMRS's gripper; (3) during post-assembly verification, SSVIS indicates that the white seal is too cocked in the bushing for the assembly to proceed.

In addition, since assembly-specific errors will be difficult to anticipate while off-line programming the ITAS, error recovery must be quick and easy to program on the shop floor. Both a simple interpretive IF/THEN shell as well as a graphical, expert-system knowledge-base tool [11] will meet the requirements for preprogrammed recovery procedures; however, an IF/THEN shell is the baseline design.

Software Architecture--Given the overall SC requirements implied above, the baseline software architecture shown in Figure 7 was established. Software modularity and concurrency are facilitated by a multitasking and/or multiprocessor environment with synchronized interprocess communications (i.e., a data exchange manager).

SC Platform Selection Criteria--The following SC hardware/software selection criteria were established. The SC *must have:* (1) real-time, multitasking operating system; (2) C language (or equivalent) development capability; (3) fast interprocess message exchange. The SC *should have*: (1) existing software applicable to ITA; (2) mature platform, near-term availability, good vendor support, reasonable cost; (3) intelligent I/O capabilities; (4) 68020 CPU (or equivalent) and 68881 math coprocessor (or equivalent); (5)

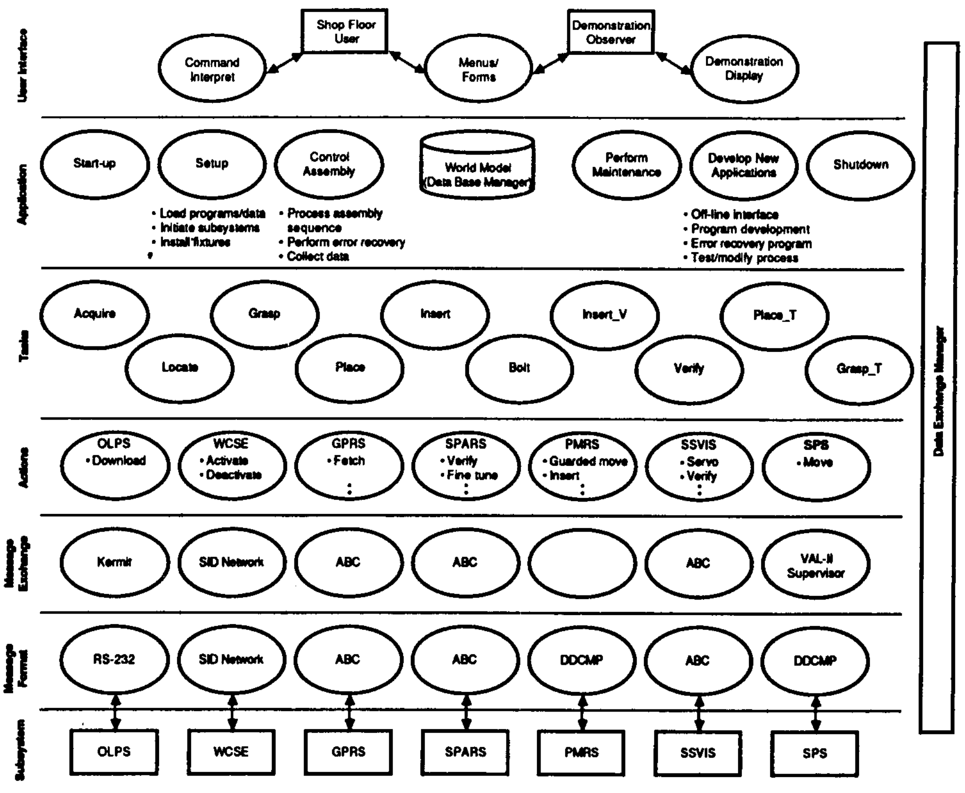


Figure 7. System Controller Baseline Software Architecture

multiuser development capability; (6) 2- to 4-MB (minimum) physical memory; (7) 60-MB (minimum) hard disk.

These criteria were then weighted and used to evaluate candidates as objectively as possible. The selected platform consists of a commercially available area controller [12] coupled with an internally developed communication gateway processor.

Global Part Recognition Subsystem (GPRS)

FETCH Process--This process provides the main functionality. For each part in the "short list of parts," a process, "Determine Acquisition Strategy," manages the following processing steps: (1) acquire imagery appropriate for the part; (2) detect features required to recognize the part; (3) build hypotheses of feature clusters that could be the part and rank them; (4) verify, in ranked order, each hypothesis to select one, if any, that is the part; if a hypothesis is probably the part, but requires verification by SPARS, label the part "quasi- found;" (5) if the part is found, capture that knowledge (to determine, if necessary, whether this part is occluding another part) and remove this part's features from further consideration; (6) if the part is found, try to find an unobstructed grasp access using preferred tool(s).

Feature Detection Process--Based on the Phase I GPRS real-time implementation study, as much of this process as possible should be implemented on an array processor. Thus, the process was decomposed sufficiently to isolate the first primitive processes for implementation; namely, the Difference of Gaussians edge detector. The resulting pseudo-code primitive process specifications have been implemented on the array processor to reduce risks with the Phase II hardware (see below).

Recognition and Grasping Algorithms--Using the Phase I GPRS functional performance results based on processing imagery from 25 tray dumps, performance and/or algorithm deficiencies were analyzed to identify any problem areas requiring immediate attention. Problem areas appear mainly in (1) the hypothesis verification step of the recognition process for small, similar-looking parts, and (2) grasp determination performance, particularly for parts with highly specular surfaces. Otherwise, the baseline algorithm suite appears to be adequate to meet requirements. Since all sensors will be new in Phase II, no further algorithm efforts (other than optimization analyses) were deemed necessary until the algorithms could be tested with data from the new sensors.

Non-Real-Time Phase II Hardware--Funding constraints preclude implementation of real-time vision subsystems in Phase II. However, functional conformance to requirements must be demonstrated within a subjectively "reasonable" time line of 16 to 25 minutes to assemble the microswitch (versus 5.4 minutes in real time), while real-time conformance will be demonstrated using a simulation of the real-time system design. As shown in Figure 8, components were selected and interface details were resolved to configure the Phase II hardware platform.

Close-Range Part Alignment and Recognition Subsystem (CRPARS)

The design activities established that a Close-Range Part Alignment and Recognition Subsystem can be developed that meets the requirements of both the Sparse Part Acquisition and Recognition Subsystem (SPARS) and the Sparse Servoing, Verification, and Inspection Subsystem (SSVIS). This represents a major simplification of the system design and development since two copies of CRPARS, rather than two different subsystems, need to be built and tested.

CRPARS Sensor Redesign--Since redesign of the Phase I prototype sensor was required to miniaturize and ruggedize it for mounting on the end effector of the high-speed PMRS, a review of the Phase II SPARS/SSVIS requirements and Phase I experiences

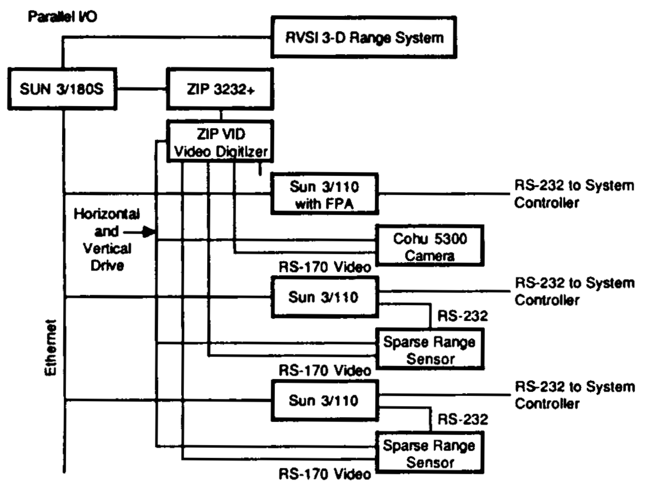


Figure 8. Phase II Vision Hardware Platform

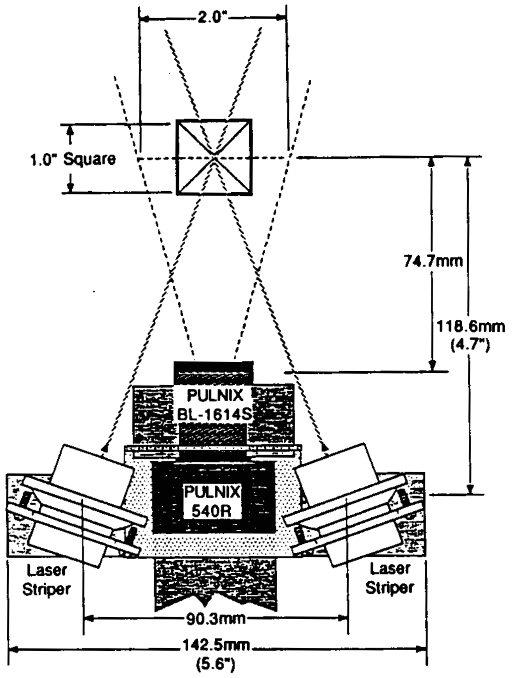


Figure 9. CRPARS Sensor Design with Two Light Stripes

was performed to design the CRPARS sensor. A 1-cubic-inch field of view is required to accommodate off-line programming and fixture placement errors–difficult to achieve with a small number of light stripes. Moreover, Phase I experiments showed problems with (1) obtaining repeatable light-stripe data, especially from specular surfaces; (2) interpreting the data from complex scenes; and (3) training and tuning light-stripe recognition models. Solutions to these problems are to (1) augment light-stripe data with binary or gray-scale data for verification and rough pose measurements; (2) add two horizontal light stripes to the two vertical stripes for SSVIS assembly verification and interpart alignment tasks; (3) use binary or gray-scale data to reposition the CRPARS sensor when light-stripe recognition has failed but when 2-D conditions look promising enough to warrant another verification attempt.

Three of the many candidates were evaluated; namely, stereo gray scale, stereo gray scale with a single light stripe, and monocular gray scale with two to four light stripes. Selection criteria included hardware/software maturity, illumination needs, size/weight, training and calibration complexity, and speed. Monocular gray scale with two to four light stripes (two for SPARS, four for SSVIS) was chosen; stereo gray scale with a single light stripe was a close second choice, but maturity based on Phase I prevailed. A component selection study resulted in the design shown in Figure 9.

CRPARS Algorithms--Algorithms for in-process monitoring and verification of expected conditions were analyzed and compared to the GPRS recognition process to maximize commonality between the two subsystems. Algorithms to measure object pose must be implemented on an array processor to meet real-time requirements. Thus, they were decomposed and analyzed sufficiently to verify that they could be implemented as planned on the selected Phase II array processor (see Figure 8).

Part Manipulation Robot Subsystem

The PMRS manipulator will consist of a five-axis AdeptOne™ SCARA robot. Since the PMRS end effector consists of a JR3 Corp. force-sensing wrist, a LORD Corp. instrumented, servoable gripper, and the SPARS (two-stripe CRPARS) sensor, the evolution of its physical configuration and payload is being monitored carefully. Using component weight estimates and a static torque analysis, the baseline configuration is within the robot's payload limits; its length drives the need for a 12-inch stroke on Joint 3 of PMRS (see Figure 5).

The Phase II PMRS hardware architecture shown in Figure 10 was designed and implemented. The major change from Phase I was replacement of the Weitek Corp. array processor with a more easily programmable and lower cost Motorola 68020 CPU/68881 math coprocessor single-board computer. A joint interface board for the fifth axis was also added. The LORD Corp. gripper has its own controller, which communicates with the Adept robot controller over an RS-232 line. The Phase I force servo control was ported to the Phase II hardware to obtain a 250-Hz force servo rate to mesh with the 500-Hz joint servo rate. After converting the force matrix multiplication from floating point to integer computation, implementing the forward arm solution in VAL II, and extending control to five axes, force servo processing requires 3 out of the available 4 milliseconds.

The Phase II system performs comparably to the Phase I system with some additional features. Five-axis force servoing allows specification of translation stiffnesses in cartesian tool (end-effector) coordinates, referencing of orientation stiffnesses to individual joints (which seems appropriate for a SCARA robot), and implementation of a true software remote-center-compliance (RCC). Joint limit stops, which gracefully pull the arm back into the active region when the offending force is removed, have replaced the panic shutdown monitor from Phase I; while performance is less than ideal, it is computationally efficient. Real-time gravity compensation, which tracks the movement of Joint 5, subtracts sensor bias and payload weight to leave contact forces; an automatic calibration routine takes measurements at several points and computes parameters using least squares.

Sensor Positioning Subsystem (SPS)

A standard six-axis PUMA 560 from Unimation/Westinghouse appears to meet all the requirements of the SPS, including communication with the SC via its "supervisor" interface using the DDCMP protocol.

Work Cell Support Elements (WCSE)

The WCSE's primary functions are to provide the tools and fixtures necessary to assist the one-handed PMRS assembly operations and to provide the mechanical and structural interface for portions of the other subsystems. Attention has focused on using the detailed

ORIGINAL PAGE IS
OF POOR QUALITY

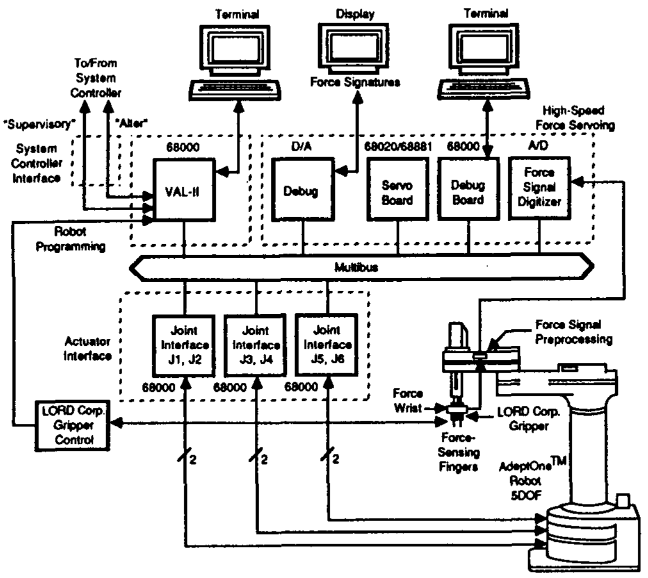


Figure 10. PMRS Detailed Hardware Architecture

assembly scenario for the 1EN1 microswitch to develop preliminary fixture and tooling prototypes for experimentation with the PMRS robot when it becomes available. The design goal is to make these as simple and multipurpose as possible.

Off-Line Programming Subsystem (OLPS)

The OLPS must provide the ability to perform system programming, program validation, program downloading and data base generation interactively for the on-line segments of the ITAS. In the Phase II ITAS, the programmer interface will not be a single one; rather, there will be one interface for the SC, one for the vision subsystems, and one for the robots (see Figure 2). For the SC, the OLPS must provide, as a minimum, the same capability as the "Develop New Applications" process does for the shop floor user. Rather than positioning the robots in position or force-teach modes, however, the simulated robots are used where the baseline implementation assumes use of the CIMStation simulation package. The results of preliminary off-line programming tests for end-to-end error buildup along with inspection of the VAL code generated by the translators were deemed adequate to keep CIMStation as the baseline. For the vision subsystems, the OLPS will consist of a copy of the shop floor interactive training processes developed for GPRS and CRPARS. Future systems would either have copies of the shop floor sensors or, ideally, synthesized training imagery from a CAD-based geometric modeler and sensor simulator. The baseline PMRS and SPS designs assume that suites of data-driven primitives can be developed such that a variety of assembly tasks can be performed without writing new primitives. In the event that this is not a valid assumption, assembly-specific routines will be developed on CIMStation, translated to the appropriate robot controller's language, and downloaded.

CONCLUSIONS

The process of engineering emerging technologies into a functional demonstration system is essential for rapidly maturing experimental implementations, validating their efficacy and, ultimately, transferring useful technology and experience. Application of systems engineering tools and techniques is an efficient approach when carefully tailored to the research nature of such projects.

ACKNOWLEDGEMENTS

This work is funded by Air Force contract No. F33615-82-C-5092 under project managers, Mr. T. Brandewie (AFWAL) and Dr. R. Rosenfeld (DARPA). The author gratefully acknowledges the contributions of many individuals at Honeywell and Adept Technology, including D. Waters, R. Bell, R. Blair, R. Graber, M. Harless, T. Hill, R. Jacobson, T. Keefe, D. Larkin, J. Mannering, J. Maples, G. Marschall, V. Nibbe, C. Raymond and B. Westover.

REFERENCES

1. Levinthal, E. C., "DARPA Program--Intelligent Task Automation," *Robotics and Industrial Inspection, Proceedings of SPIE, The International Society for Optical Engineering*, San Diego, CA, Vol. 360, August 24-27, 1982, pp. 32-38.

2. Rosenfeld, R., "Intelligent Task Automation Manufacturing Systems," *Proceedings, Second Annual Conference on Robotics and Factories of the Future*, Springer-Verlag, San Diego, CA, July 28-31, 1987.

3. Reimann, Walter, "Air Force Assembly Thrust," *Proceedings, The 1985 United States Air Force Computer Integrated Manufacturing*, Dallas, TX, April 9-12, 1985.

4. Benton, R. C., "Enabling Technology for Flexible Assembly Systems," *SAMPE Journal*, January/February 1985, Vol. 21-1, pp. 25-33.

5. Benton, R. C. and Waters, D. E., "Intelligent Task Automation," *Proceedings, The 1985 United States Air Force Computer Integrated Manufacturing*, Dallas, TX, April 9-12, 1985, pp. 195-214.

6. Honeywell Inc., "Intelligent Task Automation--Phase I," Volumes I, II, and III, Report No. AFWAL-TR-86-4122, Air Force Wright Aeronautical Laboratories, AFWAL/MLTC, December 1986.

7. Hatley, Derek J., "The Use of Structured Methods in the Development of Large, Software-Based Avionics Systems," Paper No. 84-2595, American Institute of Aeronautics and Astronautics, 1984.

8. Interactive Development Environments, "Software through Pictures™ Reference Manual," 150 Fourth Street, Suite 210, San Francisco, CA 94103.

9. Craig, John J., "Issues in the Design of Off-Line Programming Systems," Proceedings of the Fourth International Symposium on Robotics Research Conference, Santa Cruz, CA, August 1987.

10. Smith, Sidney L., and Mosier, Jane N., "Design Guidelines for User-System Interface to Software," The Mitre Corporation, Project No. 522A, September 1984.

11. Hutchins, B.; Wolff, A.; Cochran, E.; and Ludlow, P., "Design of a User Interface for Automated Knowledge Acquisition," *Proceedings of the 1986 IEEE International Conference on Systems, Man and Cybernetics*, Atlanta, Georgia, 1986.

12. Adept Technology, "Assembly and Information Management System Reference Guide," Version 1.5, 150 Rose Orchard Way, San Jose, CA 95134, April 1987.