CAD-BASED OFF-LINE PROGRAMMING APPLIED TO A CLEANING AND DEBURRING WORKSTATION

Dr. Ronald Lumia
Group Leader, Intelligent Controls Group
National Bureau of Standards
Metrology Building, Room B-124
Gaithersburg, MD 20899
U.S.A.

ABSTRACT

This paper describes a research project for CAD-based off-line robot programming. The research focuses on the specific problem of programming robots in the Cleaning and Deburring Workstation in the Automated Manufacturing Research Facility (AMRF) at the National Bureau of Standards (NBS). A baseline capability is first established with a commercially available off-line programming software package. From this experience, the package will be modified to incorporate geometric reasoning, an object oriented database, and an NBS hierarchical control system. Eventually, the off-line programming techniques will produce executable programs which will be run by the NBS hierarchical control system.

INTRODUCTION

The use of off-line programming for robots is an important research issue for two very fundamental reasons. The first reason is economic. The traditional teach pendant methods for programming a robot are often impractical because the robot cannot simultaneously be used for teaching and working. These methods represent an acceptable alternative when the time required to teach the robot is a small percentage of the total time the program will be used. Many applications fit this requirement. For example, robot welding of car frames may require a full day of programming but since the program will be used for one year, there is sufficient economic justification for the method. On the other hand, these techniques are

---

This work is partially supported by funding from the Navy MANTECH Program and was prepared by U.S. Government employees as part of their official duties and is therefore a work of the U.S. Government and not subject to copyright. Equipment listings in this paper do not imply a recommendation by NBS nor that the equipment is the best for the purpose.

inappropriate for small batch manufacturing, which is a particularly important segment of manufacturing processes.

A second motivation for off-line programming is safety: personnel safety, robot safety, and environmental safety. By simulating the actions of the robot, the robot's actions can be predicted by its human operators. This could prevent situations which are dangerous for personnel when testing the program on the real robot. Also, by simulating the robot program, there is less possibility that the robot will damage itself due to inappropriate commands. Finally, the robot can cause extensive damage to other machines, jigs, etc., in its workspace. Off-line programming and simulation can often detect these errors before the catastrophe occurs.

This paper describes off-line programming with a focused goal in mind. The robots are used in a factory environment to clean and deburr workpieces which are made by machine tools. While the application area is quite specific, the intent of the project is to develop generic capabilities for off-line programming techniques which can be applied to any robot programming problem.

The paper will first describe the AMRF project in general and the Cleaning and Deburring Workstation as it relates to the application. Then, an overview of the NBS hierarchical control paradigm will be presented. This is followed by a description of the two phases of the project. Finally, an attempt is made to identify some of the research areas which need to be explored in order to realize off-line programming.

OVERVIEW OF THE AMRF

The Automated Manufacturing Research Facility (AMRF) at the National Bureau of Standards (NBS) is a small batch, metal machining shop where robots, NC machines, AGVs, and a great deal of software are integrated into a "factory of the future" [1]. The AMRF is shown in Figure 1. It consists of five workstations: the horizontal workstation , the vertical workstation , the turning workstation , the cleaning and deburring workstation, and the automatic inspection workstation. A robot cart provides the workstations with raw material and removes finished parts. A robot in each workstation feeds the machines the required parts to complete a job. The AMRF was the first deliberate attempt to tie together sensory processing, world modeling, and robot control in a generic fashion [2,3,4].

Workpieces are sent through some sequence of machine tools to remove material from the original blank workpieces. This process often results in the creation of sharp burrs which must be removed. The Cleaning and Deburring Workstation is the final production step for the workpieces. It is composed of two robots which perform three distinct operations: deburring, washing, and buffing. The robots can work either independently or they can cooperate in the execution of particular tasks. Figure 2 illustrates the layout of the Cleaning and Deburring Workstation. For more detail about this workstation see [5].

HIERARCHICAL CONTROL FUNCTIONAL ARCHITECTURE

The fundamental paradigm of the control system is shown in Figure 3. The control system architecture is a three legged hierarchy of computing modules, serviced by a communications system and a global memory. The task decomposition modules perform real-time planning and task monitoring functions; they decompose task goals both spatially and temporally. The sensory processing modules filter, correlate, detect, and integrate sensory information over both space and time in order to recognize and measure patterns, features, objects, events, and relationships in the external world. The world modeling modules answer queries, make predictions, and compute evaluation functions on the state space defined by the information stored in global memory. Global memory is a database which contains the system's best estimate of the state of the external world. The world modeling modules keep the global memory database current and consistent.

Task Decomposition Modules (Plan, Execute)

The first leg of the hierarchy consists of task decomposition modules which plan and execute the decomposition of high level goals into low level actions. Task decomposition involves both a temporal decomposition (into sequential actions along the time line) and a spatial decomposition (into concurrent actions by different subsystems). Each task decomposition module at each level of the hierarchy consists of a job assignment manager, a set of planners, and a set of executors.

World Modeling Modules (Remember, Estimate, Predict, Evaluate)

The second leg of the hierarchy consists of world modeling modules which model (i.e. remember, estimate, predict) and evaluate the state of the world. The "world model" is the system's best estimate and evaluation of the history, current state, and possible future states of the world, including the states of the system being controlled. The "world model" includes both the world modeling modules and a knowledge base stored in a global memory database where state variables, maps, lists of objects and events, and attributes of objects and events are maintained. By this definition, the world model corresponds to what is widely known throughout the artificial intelligence community as a "blackboard". The world model performs the following functions:

1. Maintain the global memory knowledge base by accepting information from the sensory system.

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

3. Answer "What if?" questions asked by the planners in the corresponding level task decomposition modules. The world modeling modules predict the results of hypothesized actions.

4. Answer "What is?" questions asked by the executors in the corresponding level task decomposition modules. The task executor can request the values of any system variable.

Sensory Processing Modules (Filter, Integrate, Detect, Measure)

The third leg of the hierarchy consists of sensory processing modules. These recognize patterns, detect events, and filter and integrate sensory information over space and time. The sensory processing modules at each level compare world model predictions with sensory observations and compute correlation and difference functions. These are integrated over time and space so as to fuse sensory information from multiple sources over extended time intervals. Newly detected or recognized events, objects, and relationships are entered by the world modeling modules into the world model global memory database, and objects or relationships perceived to no longer exist are removed. The sensory processing modules also contain functions which can compute confidence factors and probabilities of recognized events, and statistical estimates of stochastic state variable values.

Operator Interface (Control, Define Goals, Indicate Objects)

The control architecture defined here has an operator interface at each level in the hierarchy. The operator interface provides a means by which human operators can observe and supervise the system. Each level of the task decomposition hierarchy provides an interface where the human operator can assume control. The commands into any level can be derived either from the higher level task decomposition module, from the operator interface, or from some combination of the two. Using a variety of input devices such as a joystick, mouse, trackball, light pen, keyboard, voice input, etc., a human operator can enter the control hierarchy at any level, at any time of his choosing, to monitor a process, to insert information, to interrupt automatic operation and take control of the task being performed, or to apply human intelligence to sensory processing or world modeling functions.

OFF-LINE PROGRAMMING

The ultimate goal of the NBS off-line programming project is to develop the capability to write a robot program and then download it to the NBS control system. This is shown in Figure 4. The programmer develops a robot program on the off-line system and simulates it using tools such as graphic screens. Once the programmer is satisfied that the program operates correctly, the program can be downloaded to the appropriate modules in the system.

While the robot control system has six levels in the hierarchy, only three levels are task dependent. The lower three levels are robot dependent but task independent. Consequently, these lower levels can be considered to embody a generic set of capabilities which all tasks can use. The off-line programming system does not interact with these levels.

To accomplish any task, the off-line programming system must send appropriate information into the modules in the upper three levels of the hierarchy. This is a great deal more complicated than simply downloading a program since the information must be divided among levels as well as the task decomposition, world modeling, and sensory processing modules. For example, the off-line programming system will use CAD style representations of parts in its simulation. Once the desired program is developed, these CAD part representations must be downloaded into the appropriate world model module. The instructions concerning precisely what is to be done reside in the task decomposition modules. There may also be algorithms sent from the off-line programming system to the sensory processing modules. It is expected that geometric reasoning will be used in the execution of many tasks.

The off-line programming system previously described presents the long range goal. In the realization of such a system, it is only reasonable to phase the progress toward the goal. Consequently, two intermediate phases of the project will be built. The initial phase of the project is illustrated in Figure 5 where a commercially available off-line programming software tool is integrated into the Cleaning and Deburring Workstation. The off-line programming system used is the CimStation package. A CAD system (ComputerVision) is used to develop workpiece models which need deburring. The off-line programming has a model of each robot, a model of the workcell, and a graphics package to display a simulation of the task being executed. The CAD system sends IGES files to the off-line programming system which then incorporates these workpiece models into the workcell model. The programmer works in the CimStation's programming language (SIL) to create a robot program to perform a task. After the task is simulated on the graphics screen and works to the satisfaction of the programmer, it is sent to the program post processor. For the Cleaning and Deburring Workstation in the AMRF, SIL is converted into the robot language VAL for the Unimate Robots.

The next phase of the project replaces the commercial robot controller with the NBS hierarchical system as shown in Figure 6. In collaboration with another NBS division, an object oriented database will be developed which interacts with the CAD system and the world modeling modules within the NBS robot controller. This allows a much more sophisticated use of sensory processing and world modeling in the real-time control system. This implies several enhancements to the off-line programming system: a much more sophisticated program post processor to convert SIL to the information sent to each module in the NBS controller, a method to simulate the sensory processing required to support the task, a method to calibrate the sensory feedback with the actual world coordinates of the workspace, etc. These improvements will help develop a better understanding of the problem so that progress can be made toward the ultimate off-line programming system.

CONCLUSIONS

This paper describes a research program for off-line programming. The ultimate goal of the CAD-based approach to off-line programming is to develop a "seamless" method to proceed from the application concept to the application program. The major conclusion drawn from this work is that there are several central research issues which need to be addressed by the scientific community before substantial progress can be made. It is hoped that these issues can be explored during the workshop.

The issue with the most impact is data representation. This problem is manifested in several ways for CAD-based off-line programming. The first concerns the representation of the task. It is well known that the way in which data is represented can have a profound effect upon the algorithms using that data. For example, 1078÷89 can be performed by an algorithm learned in elementary school when the numbers are represented by Arabic numerals. Suppose the numbers were represented with Roman numerals. The "congruence" between the algorithm and the data has been upset in spite of the fact that there is no change of information content. In the same way, the representation of the task in the world model modules and the algorithms which execute the task in the task decomposition modules are inexorably linked. Unfortunately, there is little theory on what a task means let alone how to represent it. There is the "gut" feeling, for example, that all assembly tasks are similar in some way but there is no clear direction on how to capitalize on this potential similarity.

A similar problem in data representation concerns the CAD representation of parts. The vast majority of CAD representations fall into either constructive solid geometry (CSG) or boundary representation (B-rep) approaches. Neither is suitable for all of the algorithms required in a robot control system [6]. Perhaps the concept of the object oriented database [7] will be useful to link together several object representations in an efficient structure so that the proper information is available to all algorithms requiring the data.

Another area of interest is geometric reasoning. Algorithms need to be developed which use world modeling data representations to automatically create reasonable sequences of actions for tasks such as assembly. This is especially important at the instant where the robot changes from free space motion to motions in contact with the environment.

It should be noted that the problem of choosing an appropriate robot language has in one sense been finessed. After the initial phase where VAL is used, the project will focus on integrating the off-line programming with the NBS hierarchy directly without the use of any commercially available robot language. This stems from the fact that the robot languages which are currently available often suffer from the melange of constructs from different hierarchical levels as well as from the hierarchies inherent in the task decomposition, world modeling, and sensory processing modules. As a result, these languages do not fit naturally in the NBS hierarchical control system paradigm. Consequently, the off-line programming system must be able to convert the representation of the task into the information required by each module. This process could be considered analogous to a compiler. In any case, a substantial amount of effort is required in the realization of this goal.

It can be argued that a large part of the incompatibility between robots and robot languages is the direct result of a lack of standards. The NASREM [8] functional architecture, which describes a generic hierarchically organized robot control strategy of telerobot, is an attempt toward developing such a standard.

REFERENCES

[1] J.A. Simpson, R.J. Hocken, J.S. Albus. "The Automated Manufacturing Research Facility of the National Bureau of Standards," Journal of Manufacturing Systems, 1, 1, 1982, 17.

[2] R. Lumia, "Representing Solids for a Real-Time Robot Sensory System," Proc. Prolamat 1985, Paris, June 1985.

[3] M.O. Shneier, E.W. Kent, P. Mansbach, "Representing Workspace and Model Knowledge for a Robot with Mobile Sensors," Proc. 7th Int. Conf. Pattern Recognition, 1984, p. 199.

[4] M.O. Shneier, R. Lumia, E.W. Kent, "Model Based Strategies for High-Level Robot Vision," CVGIP, Vol. 33, 1986, p. 293.

[5] R.J. Norcross, "A Control Structure for Multi-Tasking Workstations," Proc. IEEE Conf on Robotics & Automation, Philadelphia, PA. 1988.

[6] R. Lumia, "Representing Solids for a Real-Time Robot Sensory System," Proc. Prolamat 1985, Paris, June 1985.

[7] G. Copeland, D. Maier, "Making Smalltalk a Database System", Sigmod 84, vol. 14, 2, p. 316.

[8] J.S. Albus, R. Lumia, H.G. McCain, "NASA/NBS Standard Reference Model For Telerobot Control System Architecture (NASREM)," NBS Technical Note 1235, Also available as NASA document SS-GSFC-0027.

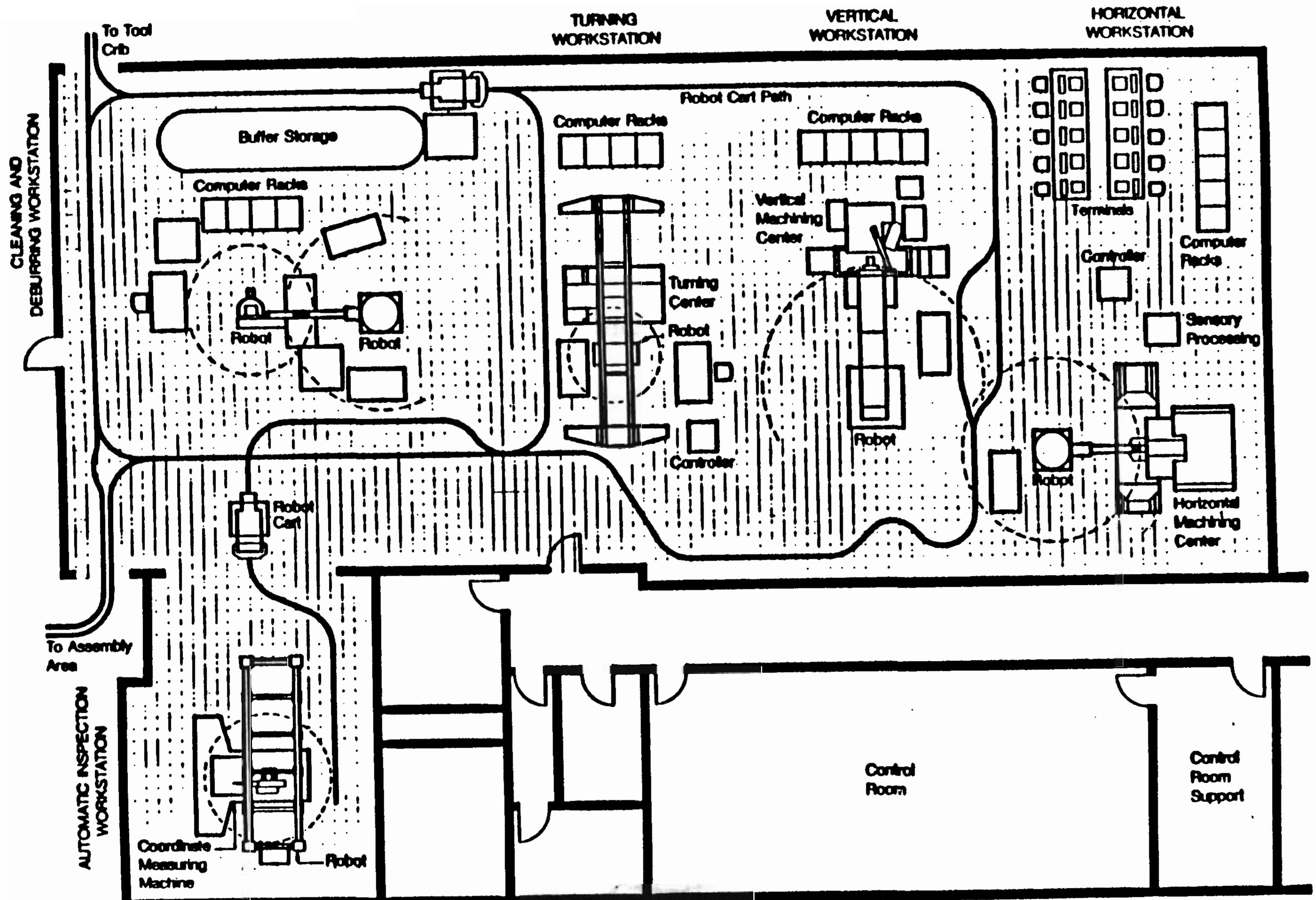

FIGURE 1 — AUTOMATED MANUFACTURING RESEARCH FACILITY

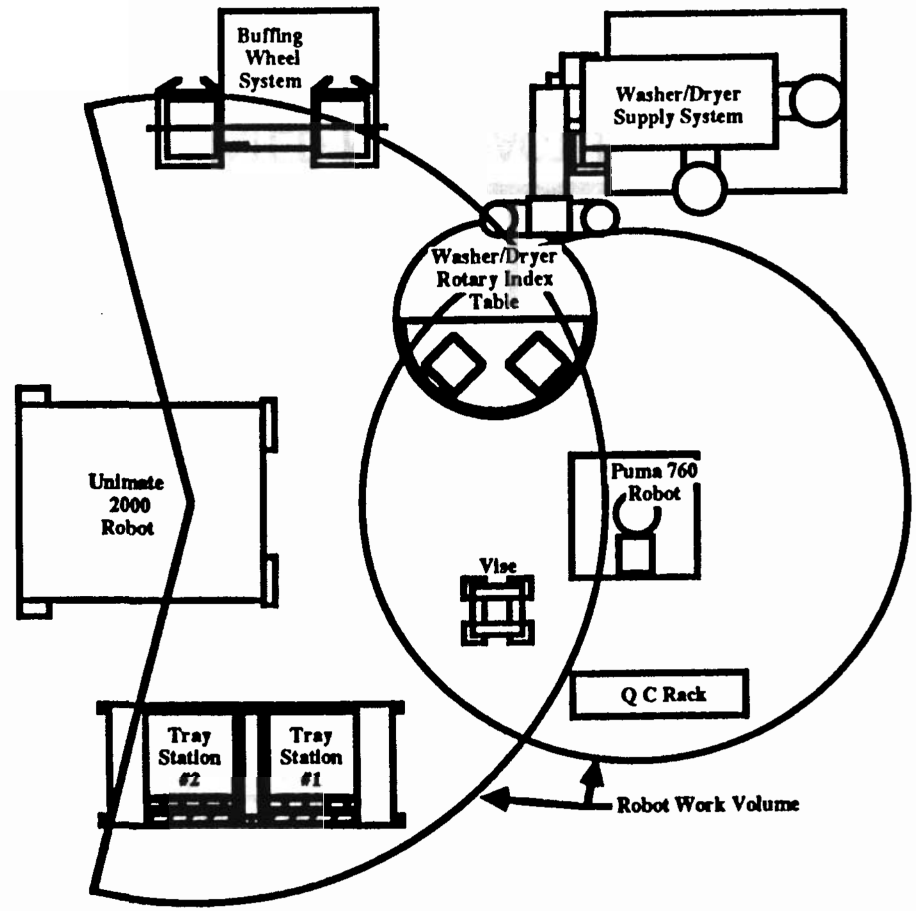
Buffing Wheel System
Washer/Dryer Supply System
Washer/Dryer Rotary Index Table
Unimate 2000 Robot
Puma 760 Robot
Vise
Q C Rack
Tray Station #2
Tray Station #1
Robot Work Volume

Figure 2

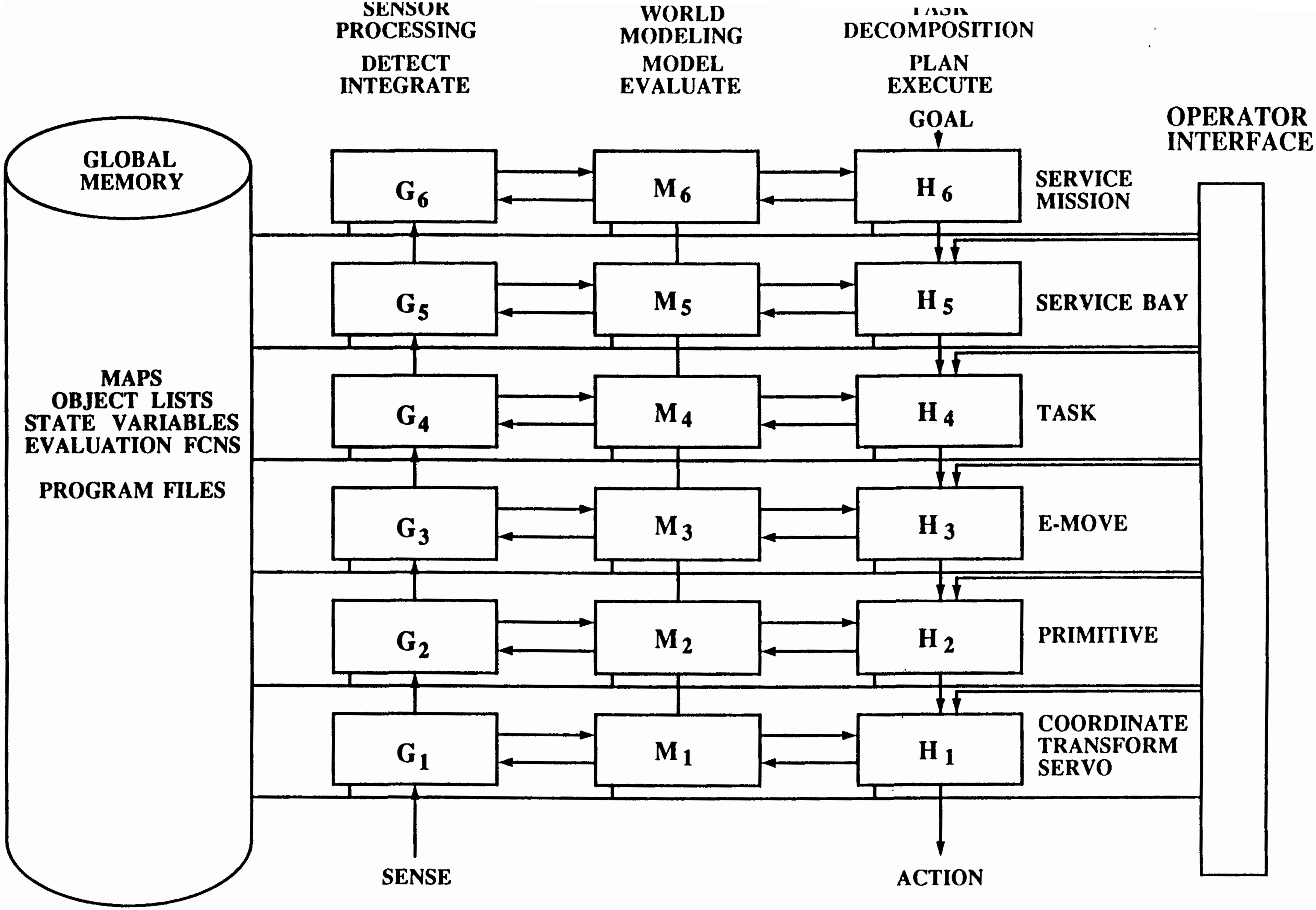

FIGURE 3 - The NBS Hierarchical Control System Architecture.

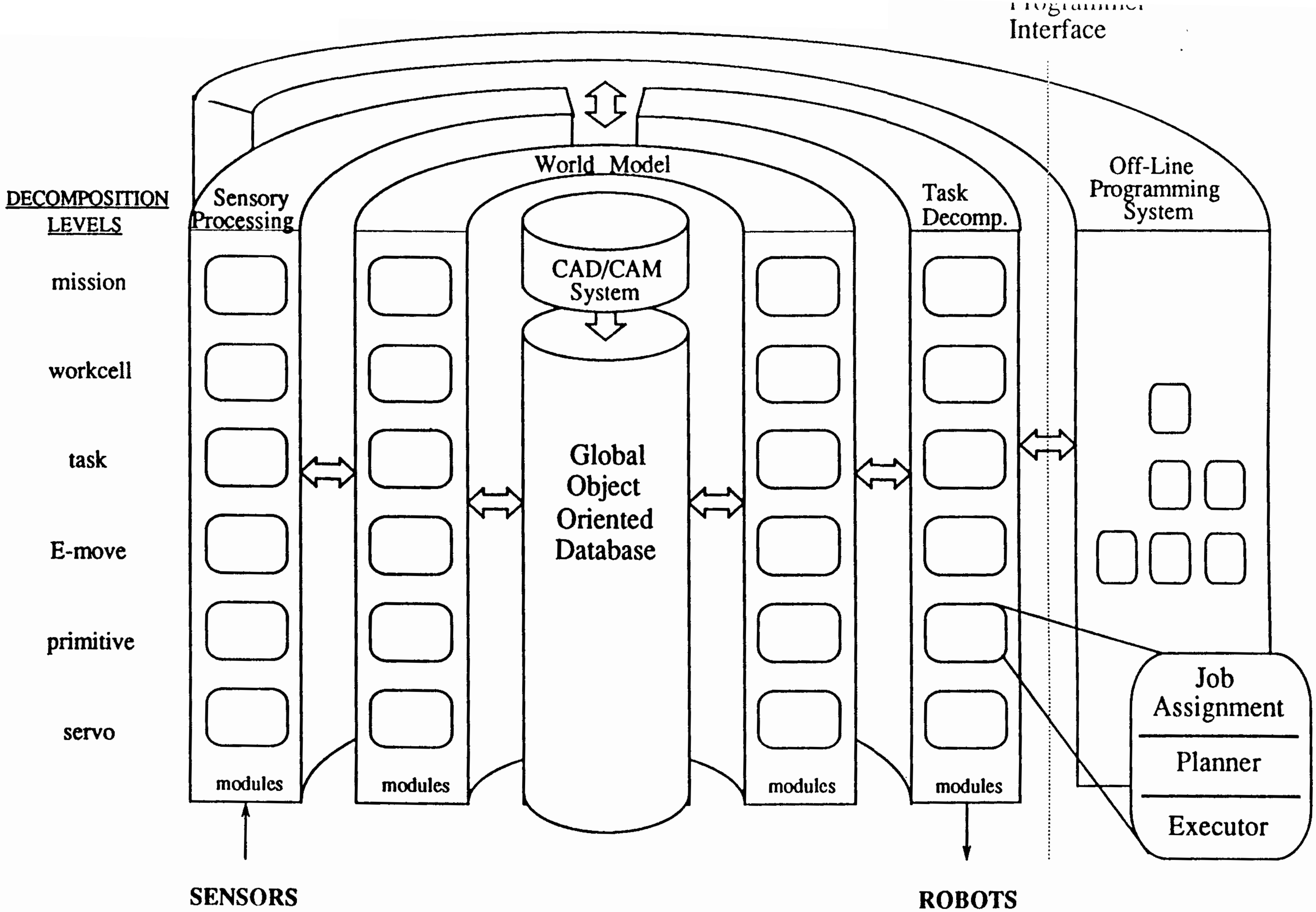

FIGURE 4 - Implementation of an Off-Line Programming system in a hierarchical control structure.

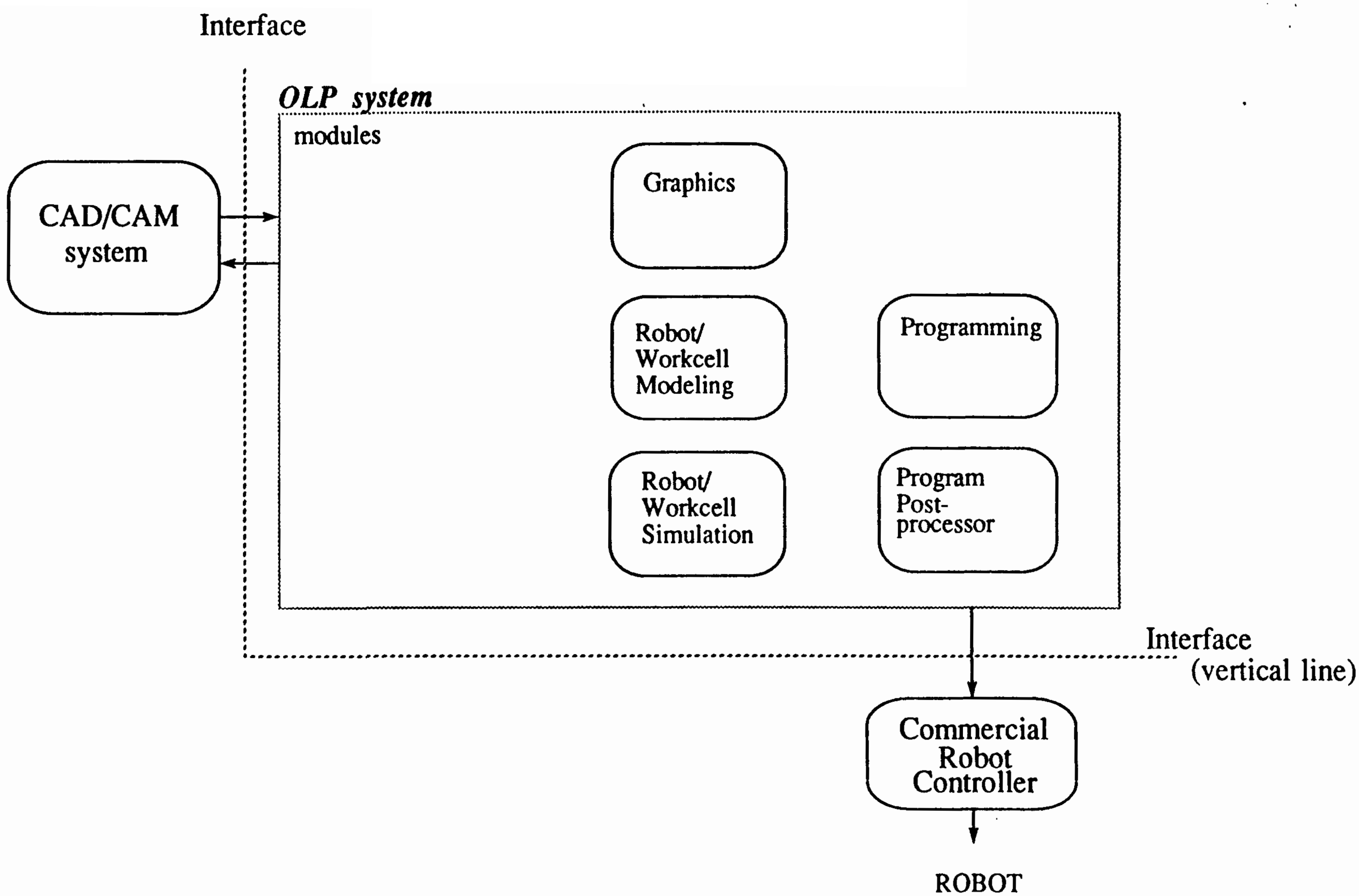

FIGURE 5 - Initial implementation of the Off-Line Programming (OLP) system

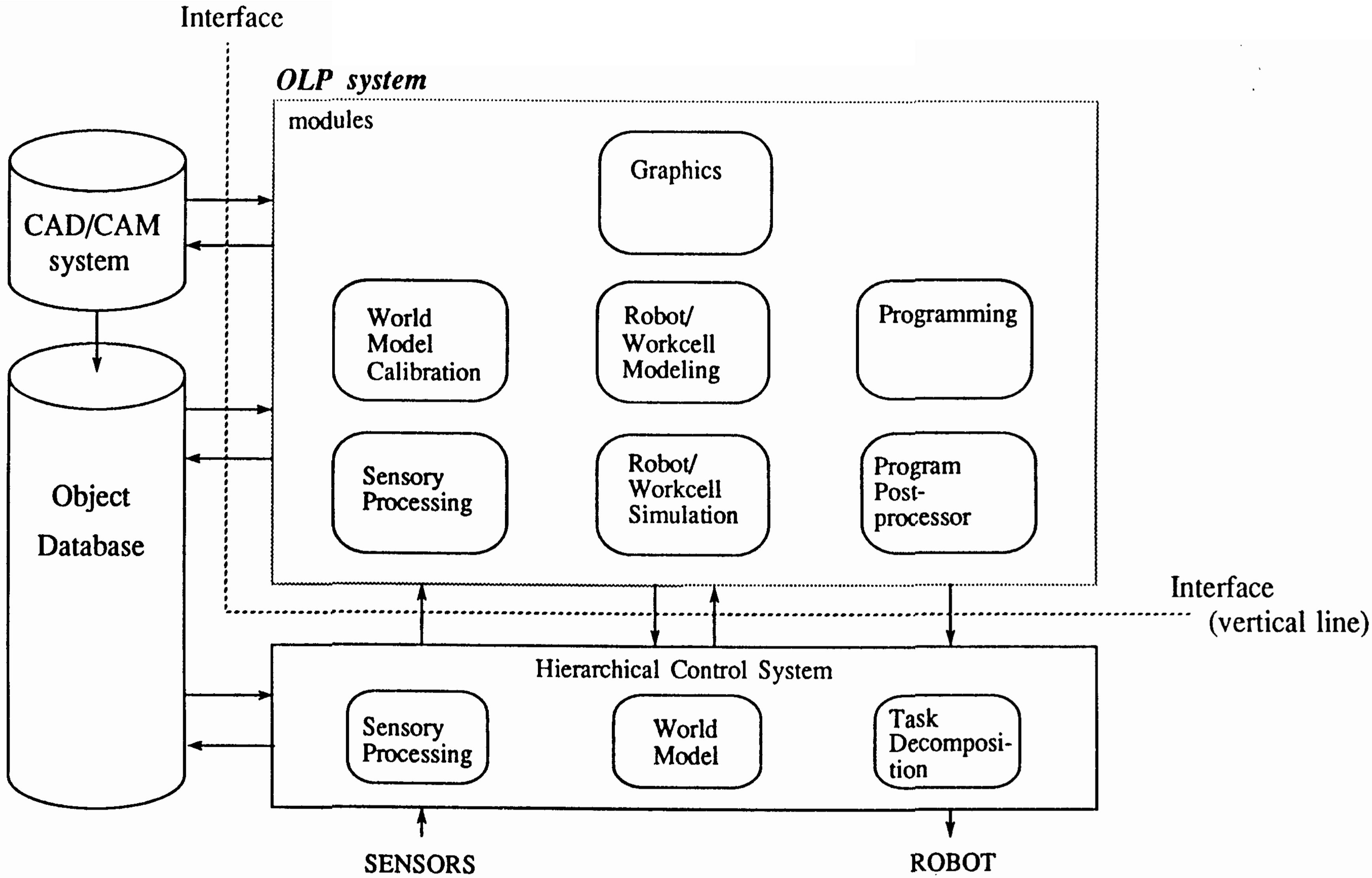

FIGURE 6 - Final implementation of the Off-Line Programming (OLP) system