The Role of Off-Line Robot Programming in Hierarchical Control

Nicholas Tarnoff and Ronald Lumia
Robot Systems Division, National Bureau of Standards
Gaithersburg, MD. 20899

ABSTRACT

This paper discusses the integration and use of an Off-Line Programming (OLP) system that is being implemented at the National Bureau of Standards (NBS) Automated Manufacturing Research Facility (AMRF). The OLP project will serve as an environment where new OLP capabilities are developed and integrated in an effort to build a complete hierarchical robot manipulation system. The two phase project first involves making a commercially available OLP system operational and integrated with a CAD/CAM system. In phase two, the system will become a development tool for feature based (object oriented) task level robot programming with the support of a global object oriented database, real time sensory capabilities, and a Hierarchical Control System (HCS).

I. INTRODUCTION

Off-Line Programming (OLP) systems are being used by industry to generate robot programs without the use of the actual robot. The primary goals of these systems are two fold. Safety, one of the important goals, is achieved by minimizing physical interaction between the operator and a powered robot. The second goal is to minimize robot down time which ordinarily occurs when programs are generated using the teach pendant method.

OLP systems enable graphical simulation of the robot, workcell, and entire factory floors for use in program testing. Such simulation systems encapsulate many aspects of automated manufacturing such as process planning, parts database management and robot specific control (downloading and executing programs). OLP systems, however, are often developed with loosely defined or no comprehensive system architecture in mind. The resulting systems lack flexibility and functional consistency. Figure 1 on the next page is the primary concept for the NBS hierarchical robot control structure that we propose to use [1]. The paradigm in Figure 1 is the basis for Figure 2 . Figure 2 shows how an OLP system is implemented as part of a comprehensive robot manipulation system. This method of implementing an OLP system focuses attention on integra-

Equipment listings in this paper do not imply a recommendation by NBS nor that the equipment is the best for the purpose.

SOURCE: 2nd International Symposium on Robotics and Manufacturing Research, Education and Aoolications. Nov.. 1988. Alburouerque. NM USA.

tion and modularity as a means of achieving human safety, and increased returns on capital investments.

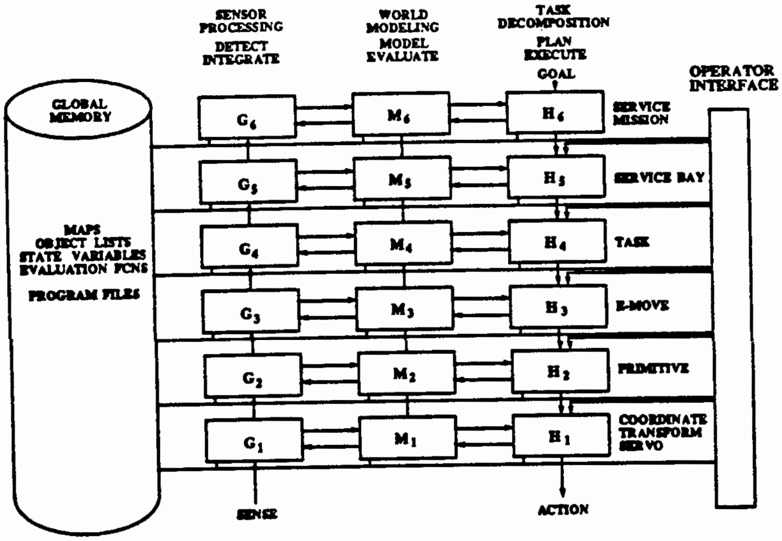

FIG. 1 - The NBS Hierarchical Control System Architecture [1].

The application area for this paper involves a cleaning and deburring workcell at the AMRF [2,3]. An OLP system will be implemented at this workcell in two phases. The first phase of the project is intended to make a baseline OLP system operational. This system will provide the basics for interactive robot programming , CAD/CAM data transfer, and program simulation. The second phase of the project will focus on developing a task level robot programming language. Applied research of such a language requires the support of a database and the fusion of sensory processing capabilities. A detailed description of the actual workcell is relegated to the reference section. The application area is very specific but the intent of the project is to develop generic capabilities for OLP which can be applied to any robot programming task [4].

We begin by explaining the functionality of the NBS Hierarchical Control Structure. In section II the implementation plans for an OLP system are presented, followed by a discussion of the projects ultimate specific goal in the area of task level programming. Lastly, critical research issues addressed by this OLP project are summarized.

II. HIERARCHICAL CONTROL

The fundamental paradigm for the hierarchical control structure shown in Figure 1 is explained in detail in [1], but a brief overview of the control structure is provided here for completeness. Figure 2 shows the NBS hierarchical control structure as having six decomposition levels (Mission, Service Bay, Task, Elementary-Move, Primitive and Servo) and three vertical hierarchies (Sensory Processing, World Modeling and Task Decomposition). Communication is possible between any two modules in the structure by way of the global database.

The world model modules facilitate the transfer of data to and from the database thus separating Sensory Processing and Task Decomposition. Data communication is primarily horizontal and much more voluminous than the primarily vertical communication of commands and status feedback. Successive decompositions of tasks across both time and space result in a tree structure of task decomposition modules. Each task decomposition module assigns jobs, plans and executes. Planning is more active at the higher task decomposition levels while modules at the lower levels do more process execution.

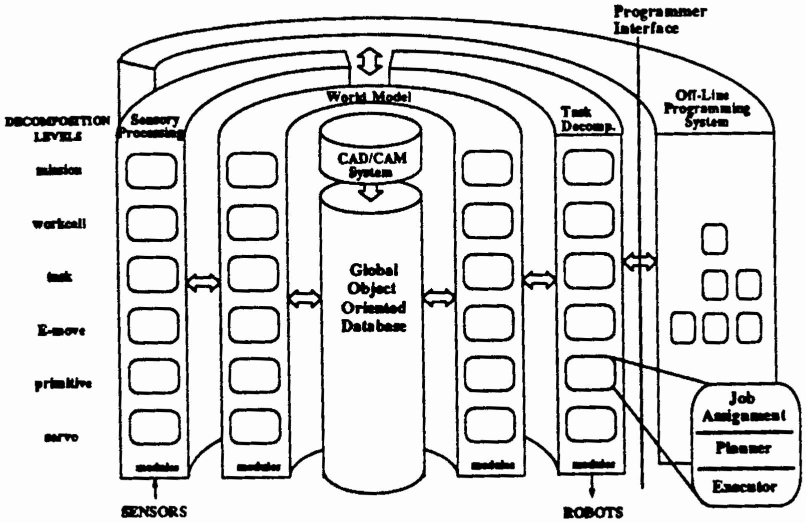

FIG. 2 - Implementation of an OLP System in a Hierarchical Control Structure.

In this hierarchical structure the OLP system is a *partial image* of the hierarchical control structure. The lower level task decomposition modules are not simulated in the OLP system since they involve task independent but robot dependent actions. The OLP system is primarily concerned with accessing and simulating task dependent but robot independent actions. The OLP system serves as an operator interface tool that is, therefore, mostly used at higher levels and almost never at the lowest task level. The OLP system's role in a hierarchical control system is evolutionary. At first, the system is a *programming* environment where desired functions of the hierarchical control system are developed (or programmed) in the OLP environment. Once the process is developed in the OLP system (to the right of the programmer interface line) it becomes a permanent part of the robot manipulation system (to the left of the programmer interface line). Once the manipulation system matures and can operate at least semi-autonomously, the OLP system becomes a safety system based on redundancy and operator supervision.

III. APPLICATION

3.1 Phase I

The initial phase of the project will essentially make the OLP system operational as seen in Figure 3. The OLP system accepts part information directly from the CAD/CAM system to create its own complete world model. Programs are then generated, and run in simulation mode before being post-processed for down-loading to a robot controller.

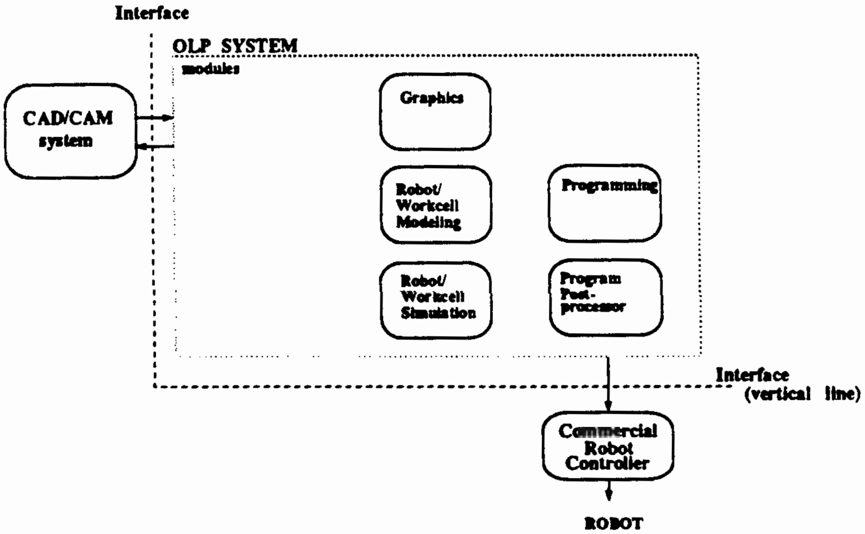

FIG. 3 - Initial implementation of the Off-Line Programming (OLP) system

As mentioned earlier the OLP system accepts part design information directly from the CAD/CAM system. The resulting "paperless" transfer of part information makes full use of a CAD/CAM system's design capabilities. It also eliminates the re-input of design information and resulting discrepancies. Most of the effort in the initial implementation of the OLP system will be spent on this interfacing of the CAD/CAM system with the OLP system.

3.2 Phase II

Phase two of the OLP system will contain the modules shown in Fig. 4. In this phase, the most important addition is the Hierarchical Control System (HCS). The Hierarchical Control System is currently being developed at NBS. It replaces the ordinary robot specific controller and provides a very strong environment for using sensory information *in real time*. Each of the levels in the Hierarchical Control System have sensory processing, world modeling, and task decomposition modules as defined by the hierarchical structure of Figure 1.

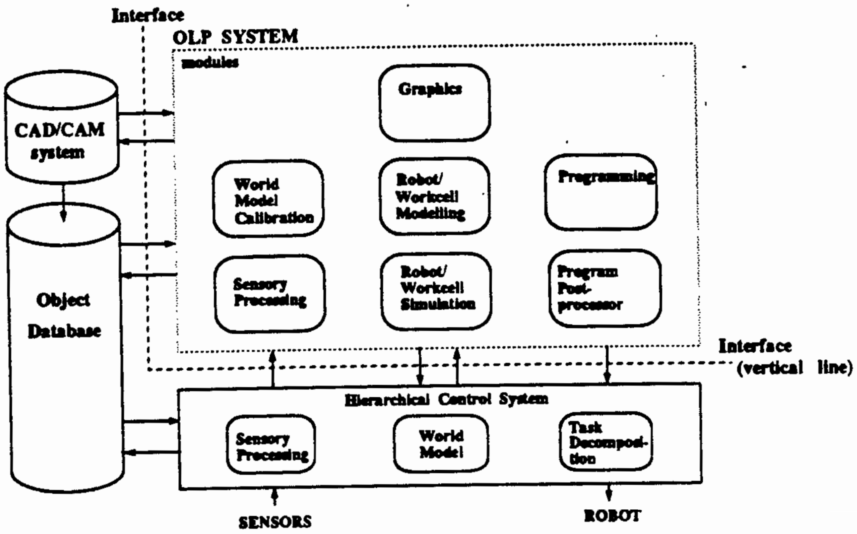

FIG. 4 - Final implementation of the Off-Line Programming (OLP) system

3.3 Development

The OLP project is divided into three areas of development: interfaces, algorithms, and structure. All three areas apply across both Phase I (short-term goals) and Phase II (long-term goals).

3.3.1 Interfaces

The integration of a CAD/CAM system with a Robot Programming and Simulation (or OLP) system is the first step in automating robot workcell programming. The kinds of information passed between CAD and OLP systems are limited by the interface standards available today [5]. The Initial Graphics Exchange Specification (IGES), which was initiated at NBS, is a widely supported first generation standard format for representing model geometry across CAD systems [6]. IGES, however, supports a limited set of primitive geometric features. IGES serves as the interface standard we use to pass simple wire-frame models from a CAD system to our commercial OLP system (SILMA's CimStation). The relatively accurate part model is then introduced in the OLP system's less detailed world model (workcell model). The world model is used to interactively generate and test robot programs off-line.

The need for a more comprehensive interface standard has prompted NBS to develop a next generation CAD/CAM interface standard called the Product Data Exchange Standard (PDES) in collaboration with the International Standards Organization's (ISO) (Standard for the Exchange of Product Model Data [STEP]) [7,8]. PDES is meant to support both the range (breadth) of information necessary for a products entire manufacturing life cycle, and the detailed (depth of) information necessary for complex and precise products. Once the "vocabulary" of PDES matures, CAD/CAM data will become available to all sectors of CIM via global object-oriented databases.

Object-Oriented (rather than relational) databases store data or groups of data as a network of *objects*. This data abstraction parallels a CIM environment much more closely than existing business oriented databases.

The support of an object oriented database, provides the valuable means of storing relationships (associativities) between features of a model, or between a feature and any other data. The creation of any drilled hole, for example, is associated with a finite number of possible diameters based on the availability of drill bits. In addition to facilitating the data exchange interface, the object oriented database also accepts data from any process (module) in the hierarchy, making the data available to the entire robot manipulation system. Thus, the state of the entire system is always current and accessible.

Emphasis in the second phase of our OLP project will shift to using CAD/CAM systems that are capable of generating the wealth of information that is now communicable. Capabilities that already exist, such as Non-Uniform Rational B-Splines (NURBS) based solid modelling (rather than wireframe) are a requirement for OLP. NURBS are a uniform and consistent way of mathematically representing all model geometry [9]. In addition, parts must be modelled as spatial volumes to support automatic collision detection, feature based manipulation of models and volumetric properties.

3.3.2 Algorithms

Initially, algorithms and features that already exist in the commercial OLP system, such as collision detection, and motion tracing will be enhanced to provide follow up procedures (post-processing) [10]. Other functions such as kinematic and dynamic emulation, and robot language translations, will be modified as the need arises or as major improvements are found. The second phase of the project will be more active in the area of algorithms. Sensor driven world model calibration will be an important goal. The usefulness of an OLP system world model relies on the accuracy of that model. Improvements in both hardware and software systems and the *open architecture* of our OLP system will enable us to add new capabilities as they are identified.

3.3.3 Structure

Modularity and an open system architecture are emphasized throughout the project. They are paramount to achieving the functionality and hierarchical structure around which the OLP system is being developed. The addition of the Hierarchical Control System (HCS) in the second phase will be a major step in implementing a modular and hierarchical OLP system. The HCS will complete the three lower control levels (Servo, Prim and E-Move) as illustrated by Figure 1. The Hierarchical Control System will enable us to experiment with sensory fusion and real-time sensory driven control. Most importantly, our HCS enhanced OLP system will now be a suitable environment for work on a task level robot programming language.

IV. TASK LEVEL, FEATURE BASED ROBOT PROGRAMMING

The robot programming languages available today do not offer the abstraction, features and tools that robot programmers find necessary [11]. Feature based (or object oriented*) robot programming is a method that addresses some of these robot programming issues. Consequently, the final goal of the project is to develop a task level object oriented robot programming language. Briefly, object oriented robot programming refers to a level of abstraction where geometric entities (surfaces, holes, etc.), rather than positional coordinates, are used to specify a robot task. Commands are generated by pointing to geometric entities on a graphics screen. A typical object oriented command is: "FOLLOW EDGE A UP TO SURFACE B" or "ALIGN SCREW C WITH THREADED HOLE D". Tasks are specified relative to the object (implicitly) rather than relative to the manipulator (explicitly). A graphical object oriented method is more similar to how humans act upon their environment and is therefore easier to use.

V. SUMMARY

The OLP system in its final form will be modular, integrated and easy to incorporate as part of a complete robot manipulation system. The OLP system will provide the powerful environment that is absolutely necessary for developing a robot language without straying towards algorithmic development.

- A CAD/CAM system will provide accurate design of part models to the OLP system via a data exchange standard.
- The Global Object Oriented Database will be a communications buffer for information concerning every module of the system.
- The Hierarchical Control System (HCS) will replace the traditionally robot specific programming languages and robot controllers. The results will be flexibility, and a powerful environment for using sensory information in real time.
- The OLP system will facilitate sensory processing (to calibrate the world model), graphic modeling of the world model, object oriented programming, and program simulation.
- The strength of the OLP system lie in the interaction between the operator and object oriented robot programming facilities.

The system described is intended for active use and will serve as a tool for continuing research in the area of autonomous and semi-autonomous robot control.

*An object oriented *robot* programming *method* does not refer to what is commonly known as object oriented programming. Object oriented *robot* programming may, however, be implemented using object oriented programming.

REFERENCES

1. J.S. Albus, H.G. McCain, and R. Lumia, "NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)" NBS Technical Note 1235: National Bureau of Standards, Gaithersburg, MD. Also available as NASA document SS-GSFC-0027, (1987).

2. 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, 17 (1982).

3. H.G. McCain, R.D. Kilmer and K.N. Murphy, "Development of a CDWS for the AMRF", SME Proceedings of the Deburring and Surface Conditioning Conference, 2, 26-43 (1985).

4. Ronald Lumia, "CAD-Based Off-Line Programming Applied to a Cleaning and Deburring Workstation", NATO Advanced Research Workshop on CAD Based Programming for Sensor Based Robots, (Il Ciocco, July 1988).

5. CAD/CIM Alert, Management Roundtable, The Manager's Guide to CAD/CAM Standards for Integration, Jonathan Linden, editor, Management Roundtable, Inc., (Chestnut Hill, Massachusetts 1986).

6. Bradford M. Smith, Joan Wellington, "Initial Graphics Exchange Specification (IGES), Version 3.0", National Bureau of Standards Document no. IR 86-3359 (April 1986).

7. Bradford M. Smith, "U.S. and International Efforts in Product Data Exchange", National Bureau of Standards, (January 1986).

8. D. Appleton Co., "Development Plan for PDES Version 1.0", D. Appleton Company, Inc., Document no. 669-87-03, Submitted to South Carolina Research Authority, (July 1987).

9. Leslie Piegl and Wayne Iller, "Curve and Surface Construction Using Rational B-Splines", Computer Aided Design, 19, 9 (November 1987).

10. Craig J. John, "Issues in the Design of Off-Line Programming Systems", Fourth International Symposium on Robotic Research, (Santa Cruz, California, August 1987).

11. Richard A. Volz, "Report of the Robot Programming Language Working Group: NATO Workshop on Robot Programming Languages", IEEE Journal of Robotics and Automation, 4, 1, (February 1988).