A GENERIC USER-LEVEL SPECIFICATION FOR OPEN-SYSTEM MACHINE CONTROLLERS

I. Greenfeld and P. K. Wright
Courant Institute of Mathematical Sciences
New York University
New York, New York

ABSTRACT

A generic, and somewhat heuristic, specification for an advanced open-system machine controller is presented. Future machine-tools will have a considerable capacity for unattended operation, through subsidiary devices and sensors. An open-system controller, based on the state-of-the-art standard computer equipment, is suggested as a solution for the control of such a diversified environment. While hardware implementation of the concept is currently being conducted, the paper presents thoughts relating to some of the central software issues involved. In particular, the paper presents a specification for two software components: An extensive graphic user's interface that encompasses all the functions of the controller, including the designer screen, the single-machine operator screen, and the multi-machine plant operator screen. And, a universal manufacturing language that enables the control of a multi machine, device and sensor environment in a concurrent way, with the application of artificial intelligence and adaptive control techniques. Both specifications are currently being gradually implemented in the MOSAIC project.

1. INTRODUCTION

Machine tools are faced today with market demands for high diversity products, and for lower prices and higher quality. Thus, small batch manufacturing, combined with extensive automation, is gaining importance as a central issue in the design and implementation of machines and machines controllers. The versatility and flexibility, which are the basic prerequisites for this type of manufacturing, may be achieved by *self-sustaining open-system* machine tools [1]. These machines are equipped with subsidiary devices and sensors needed for handling stock and parts, supervising production, and performing in-line quality control. Powerful computers and databases support design, process planning, and low level control. *Open-system controllers* are the key to this technology.

Hardware and software specification and design of an open-system controller is currently taking place as the MOSAIC project at NYU [2]. It is our intention to present here a more generic, and to a large extent - heuristic, view of open-system controllers, by way of focusing on two main software issues: the *user interface* and the *manufacturing language*.

Only a few research and development programs concerned with next generation controllers are active. The U.S. Airforce Next Generation Machine-Workstation Controller (NGC) [3] provides elaborate and rigorous specifications for such controllers, concentrating on high performance and reliability. The Intelligent Machining Workstation (IMW) [4] project concentrates on user software development implemented on current controller technology while IBM's "Configurable System for Automation Programming and Control" [5] provides a programming and real-time software and hardware environment for open-system controllers.

This paper presents an *artist's view* of the controller, from the user's perspective. We believe that this basic "picture of the world" is an essential initial step in the development of the next generation controllers. Section 2 describes the future environment of machine tools, and the open-system controllers associated with it. Section 3 presents the machine's user interface through an artist's view graphics. Section 4 presents our concept for a universal manufacturing language.

2. NEXT GENERATION CONTROLLERS AND THEIR ENVIRONMENT

Figure 1 illustrates a typical environment for a future machine tool. It features: a manufacturing machine (a milling machine in this case); a robotic hand for part, stock and fixture handling; an ancillary tools magazine used by the manipulator for miscellaneous operations such as chip clearance; a stationary and/or moveable vision system, for process supervision and part inspection; a touch-trigger probe (or a laser probe) for part inspection and localization; and a programmable fixturing system. The controller section includes: a general purpose computer; a CAD/CAM program for part design; a

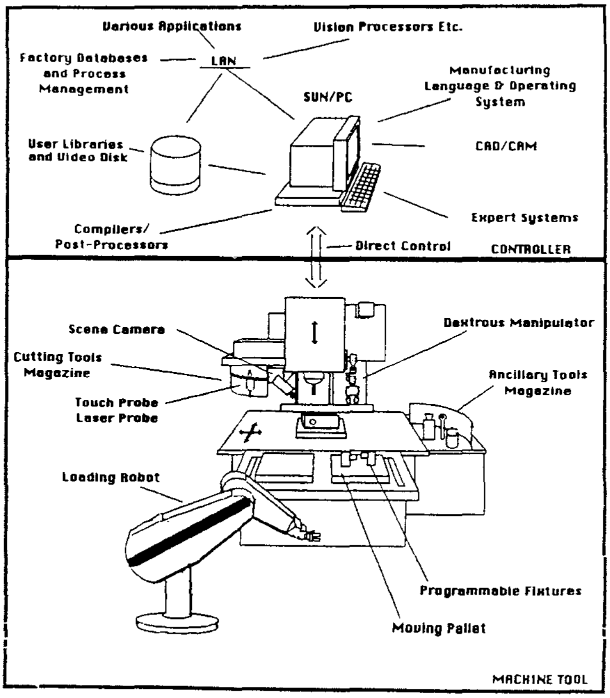

Figure 1. The Environment of Next Generation Machine Tools

manufacturing language programming environment enhanced with a manufacturing expert system; libraries of user data on various media; and a connection to local and wide area communication networks.

Figure 2 shows an example of a machine environment from the NYU - MOSAIC project. A robotic hand, shown loading a part, is mounted on the machine. A number of cameras are monitoring the process scene.

Figure 3 shows a generic architecture for an open-system controller. A few machine controllers and a plant controller are sharing a local area network (LAN). The machine controller, located close to the machine, provides the low level real-time control of motion, vision and I/O. The plant controller, which is typically remote, supervises a few machines, and provides workstation services like graphic user interface, data storage and communications.

The diversified environment of machines, devices and sensors, all controlled concurrently, requires a careful consideration of system-user interaction. An integrative approach is necessary to

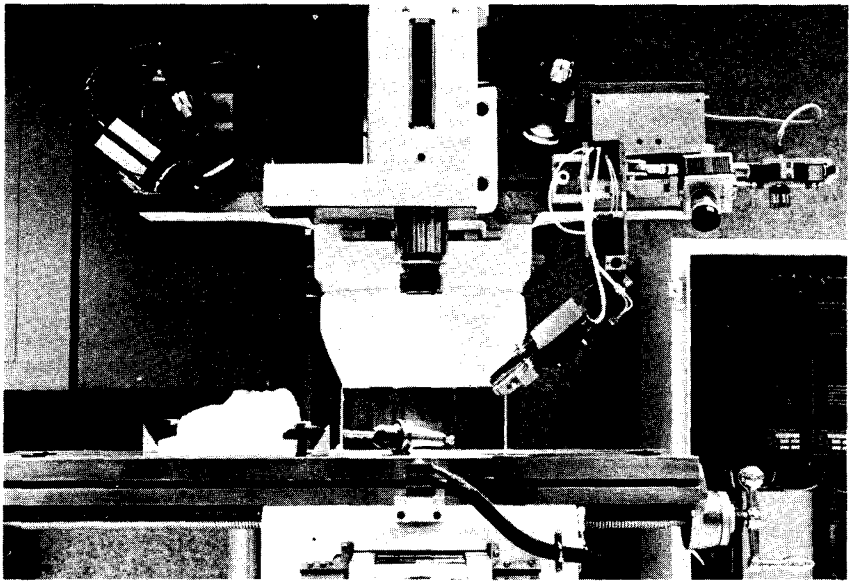

Figure 2. Devices and Sensors on a Machine-tool - The MOSAIC Project

provide the user with a clean and unified operational environment, at all levels of interaction, from the designer and planner, to the plant operator, and down to the machine operator. Through the use of existing graphic tools, as described in section 3, we are stating our view as to how the various software modules of the system should function and interact between themselves and with the user.

In addition, a universal manufacturing programming tool is needed to support this real-time multi-device environment. While many manufacturing languages exist, they are typically designed with a specific application in mind. Among them are the Automatically Programmed Tools (APT) [6] used for machining processes; IBM's Manufacturing Language (AML and AML/X) [7] for robotics, machine vision and computer aided design; automatic assembly languages like [8]; the Cell Manufacturing Language (CML) [9] which interprets existing languages relating to various applications; and Relay Ladder Logic [10] used for programming discrete operation devices. To supplement these areal-time, object-oriented manufacturing language is needed to program the various applications under a unified programming environment. This language will also serve as a universal medium for manufacturing data, and as a powerful educational tool. In the description of our concept, in section 4, we try, again, to put the emphasis on the abstract level of the concept, rather than details of language structure and syntax. The interaction with cadcam systems is stressed, as well as with expert systems, such as Machinist [11] which provides some of the human machinist's skills in process planning.

It is not our intention to invent new basic tools for computer graphics and programming, but rather, to use current software technologies to demonstrate our concepts. Graphic tools are already being used in some process control systems. A survey of applications, mainly for the energy and chemical industries, is contained in [12]. A data acquisition and control system, LabView from National Instruments [13], uses the Apple Macintosh as a workstation for programming and controlling laboratory instrumentation. Similar graphical tools are used here for the purpose of describing the concept of the user interface. Object-oriented programming languages have been around for a while, and their use is gaining popularity. Some of the more important ones are Smalltalk-80 [14], ADA [15] and C++ [16]. These can be used as building blocks for an application oriented manufacturing language.

3. THE USER INTERFACE

3.1. Overview

The open-system controller entails an extensive graphic interface. Figures 4, 5 and 6 present an artist's view of such an interface. An example from the NYU-MOSAIC project is shown in figure 7. The purpose of this presentation is twofold:

- To lay out a generic specification for future controllers, by way of describing some of their functionality.
- To demonstrate the application of well-established graphics' tools to an industrial controller.

The user interacts with the system through a conventional computer terminal, rather than through a special operator's panel. He may open a window for each application. He uses the mouse (or another pointing device) to select options by clicking on icons, menus and dialogue boxes, and to move features by click-and-drag. Thus, the conventional operator's panel is emulated by the software in a more powerful and flexible version. Three main user-interaction screens are shown:

- The Designer/Planner Screen - figure 4.
- The Machine Operator Screen - figures 5 and 7.
- The Plant Operator Screen - figure 6.

Each of these screens represents a phase in the design and production cycle, and all the applications share common data bases of parts, tools and stock and a common user interface. Many of the functions described, such as CAD/CAM, expert systems, process planning, alarm diagnostics, etc., already exist as stand-alone software packages available from various vendors. The view depicted here concentrates on the integration of these and new functions into a comprehensive design and operation environment. A description of each user screen follows.

3.2. The Designer/Planner Screen

The *Designer/Planner Screen* includes the following windows:

- *Design Window*:

This window is built around an object-based solid modelling CAD/CAM system. The solid primitives relate to physical entities, like hole, wedge, pocket, etc., rather than to geometrical units (cylinder, box, etc.). This enables the system to use the part's model for the generation of manufacturing processes, such as the automatic creation of tool paths and a process sheet, with the aid of the process planning function described later. A variety of libraries is available for the designer, including parts, tools, and processes. Tools and fixtures may be overlaid on the part's design, and manufacturing process issues considered. These libraries may be attainable from local or factory-wide databases, as well as from nationwide databases. The easy access to data and the automatic features of the system make it possible for the designer and process planner to be the same person, thus achieving higher efficiency.

- *Process Planning Window*:

This window provides an automatic process planning capability. The part design is retrieved from a CAD/CAM system, preferably in a universal part representation format like IGES or PDES. The part feature definitions and sides are extracted from the object-based solid modeller automatically, and fed into a manufacturing expert system (in this case a machining expert). The expert system prepares the process plan, including the definition of setups and the order of operations, using inherent rules. As in the design window, the same libraries are available, and tools and fixtures may be called into the design. The final process may be simulated for verification.

- *Adaptive Control Editor Window*:

This window provides a graphic editor which enables the user to define the system operational rules (or control laws), as a function of the state of the system. Physical parameters such as tool temperature, vibration, noise and torque, and input from vision systems may be used to monitor the process. An expert system analyzes the data and makes decisions relating to the process. Typical decisions may be to change feeds or gains, change tools, modify a tool path, operation abort, etc. These decisions are fed to the CAD/CAM system for regeneration of tool paths. The editor provides tools for the graphical representation of the full decision system, by using icons representing objects like gauges, parts, tools, tool paths, and decision switches.

- *Programming Window*:

This window provides a full scale programming capability , using a unified manufacturing language. The virtual setup of the machine is created by selecting menu-objects like manipulators, cameras, fixtures, etc., and placing them in the layout. Previously used layouts may be retrieved from library. Operations objects are also represented as icons like push-buttons, devices, etc. Most of the programming is done in a virtual teach-mode, whereby the programmer moves around the machine axes, manipulator gripper, parts, stock and other devices, by click-and-drag mouse operations, and selects canned operations, such as loading, cutting and inspecting, from menus. A program script is then created automatically, in any of a given selection of manufacturing languages. A fully concurrent operation of all components, though, can be achieved by a unified language only. Decision strategies, supported by the expert system, are incorporated into the program as modal statements. The final program may be simulated for verification.

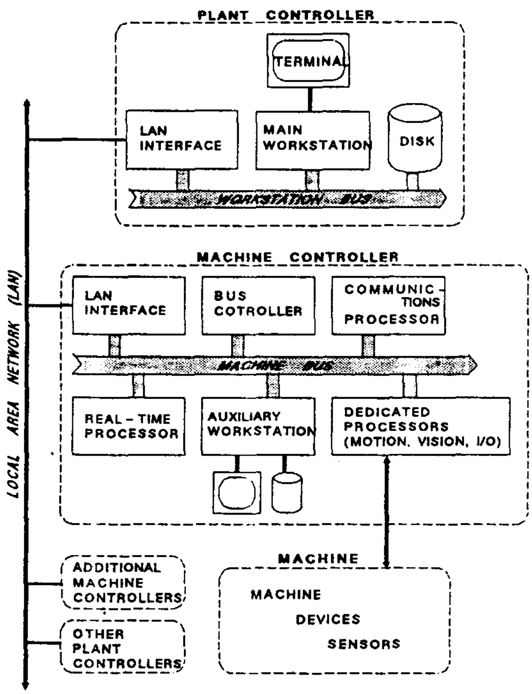

Figure 3. A Configuration for an Open-System Machine Controller

3.3. The Machine Operator Screen

The *Machine Operator Screen* includes the following windows:

• *Operations Window*:

This window provides the machine operator with a full access to all the components of the manufacturing environment. The machine setup is created by moving and manipulating fixtures, stock, cameras, etc., according to the setup plan already defined in the programming window. This operation, naturally, demands high modularity of the machine, and the availability of all the subsidiary devices and tools over a transfer line such as a conveyor belt. Cutting tools do not have to reside in a local tool magazine, but ratherneed to be able to be called up whenever needed. The operator can move and manipulate objects in a similar way to the programming window, and, in addition, he can watch the changes on the screen in real time. He can run programs and watch their progress, or operate the machine in a manual mode, by selecting options from the menus or by click-and-drag. Alarm messages are displayed as flags that point at the component at fault.

• *Tool Paths Window*:

This window shows the current progress of a tool path, along with the cutting program prepared in the programming window. The operator can verify in real time the accuracy of the current path. He may also preview on-coming tool paths or part setups, and the path decision strategy if applied.

• *Monitor Window*:

This window provides the user with a flexible monitoring interface. Any combination of gauges, in any available meter style, may be constructed by the operator and possibly saved for future use.

DESIGNER / PLANNER SCREEN

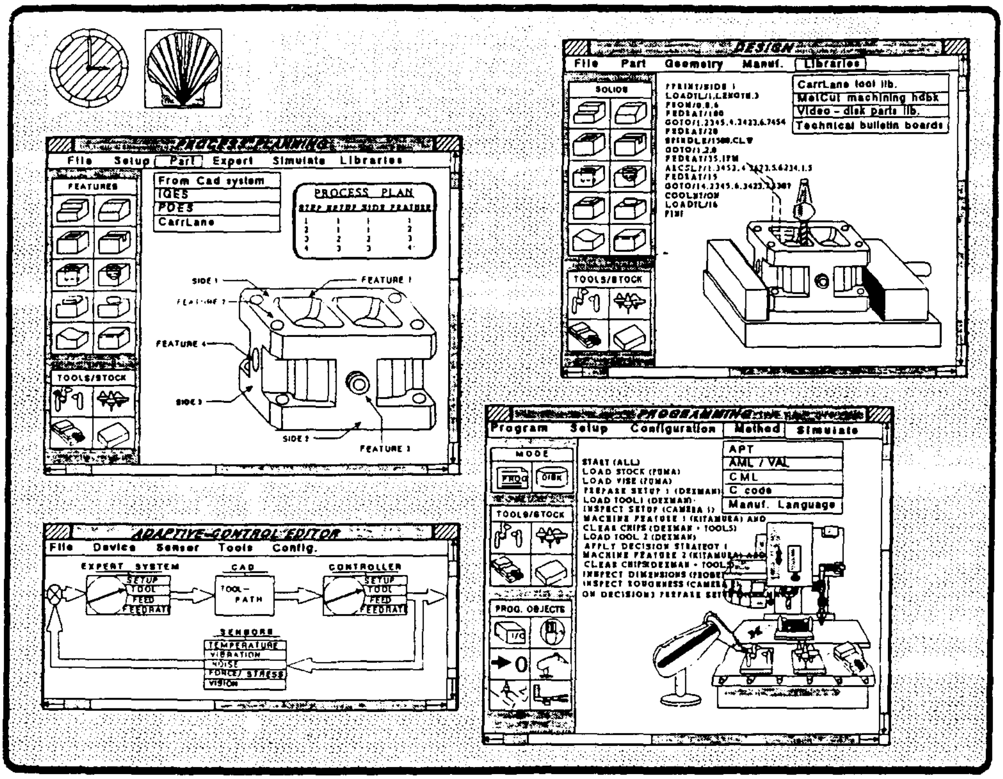

Figure 4. The User Interface - The Designer/Planner Screen

Each gauge is measuring a particular manufacturing parameter, such as axes' position and velocity, or tool's force, torque, temperature, etc.

• *Alarm Diagnostics Window*:

This window provides an alarm diagnostics facility, using a preconstructed fault tree, and expert system techniques. Cross reference of alarm flags with those presented on the operations screen is available. A few such windows may be opened simultaneously.

• *Video Window*:

This window provides a video monitor, opened as a window tool on the computer. It allows the operator to view the manufacturing scene remotely, by switching between cameras, and slewing and zooming them towards a target. A few such windows may be opened simultaneously.

MACHINE OPERATOR SCREEN

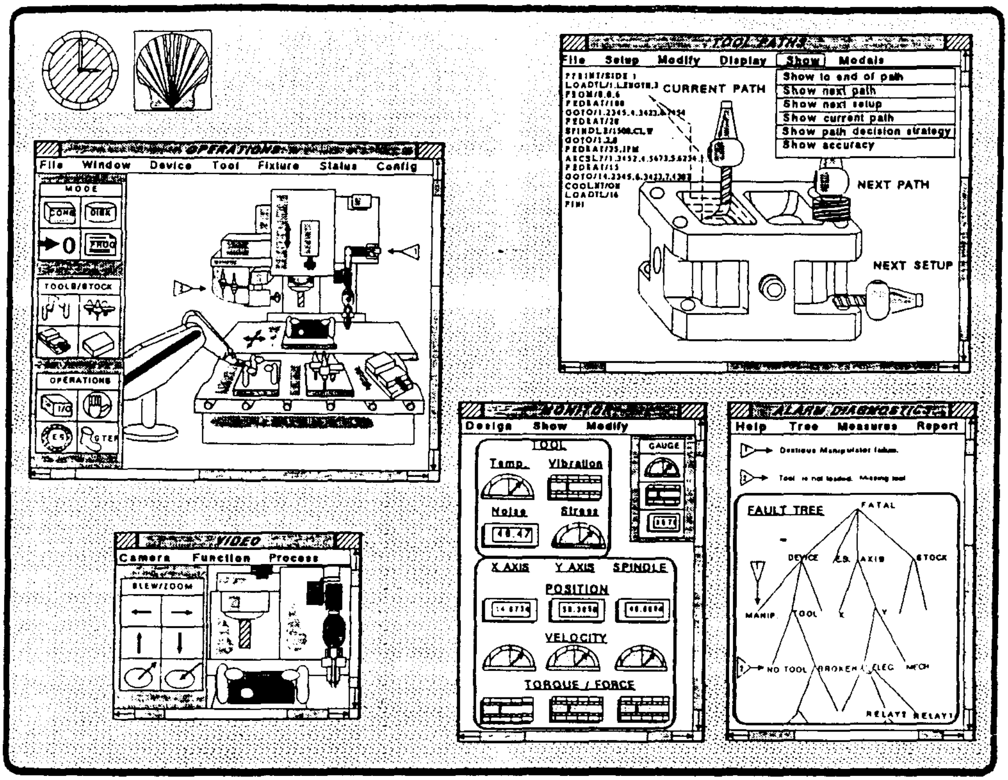

Figure 5. The User Interface - The Machine Operator Screen

3.4. The Plant Operator Screen

The *Plant Operator Screen* includes the following windows:

- *Main Operations Window*:

This window provides the plant operator with a layout of the factory floor. He can create or modify the layout, according to the physical layout, by selecting options from menus. He can move and manipulate the flow of stock and parts. He can zoom-in on a particular machine for a follow-up on the progress of the process, or for direct control in case of need.

- *Other Operations Windows*:

Any combination of machine-specific windows, such as those described before, for any machine under supervision of the plant operator, may be presented on the screen for his use.

4. THE MANUFACTURING LANGUAGE

A factory consists of a multitude of manufacturing processes, planning operations and associated databases. When human beings are confronted with this environment they "see" it with different styles of representation: an isolated metal component can be "seen" as a static solid object, onto which dimensions and tolerances can be added in the form of numerical data; a milling operation can be "seen" as a kinetic process with a rotating tool, moving table and forming chips onto which velocities, feedrates and tooling geometry can be added; a planning function can be "seen" as a process of chronological steps onto which iconic diagrams can be added with directional arrows and, possibly, feedback loops; a quality control function can be "seen" as retroactive plan combined with numerical data on the expected size of a part. In fact, different people in a factory will "see" the above operations in many different ways: a milling operation will be "seen" by the machinist in terms of speeds, feeds, tool life and surface finish but will be "seen" by a process-planner as a shape-changing function on a random machine in an arbitrary place in a manufacturing system.

PLANT OPERATOR SCREEN

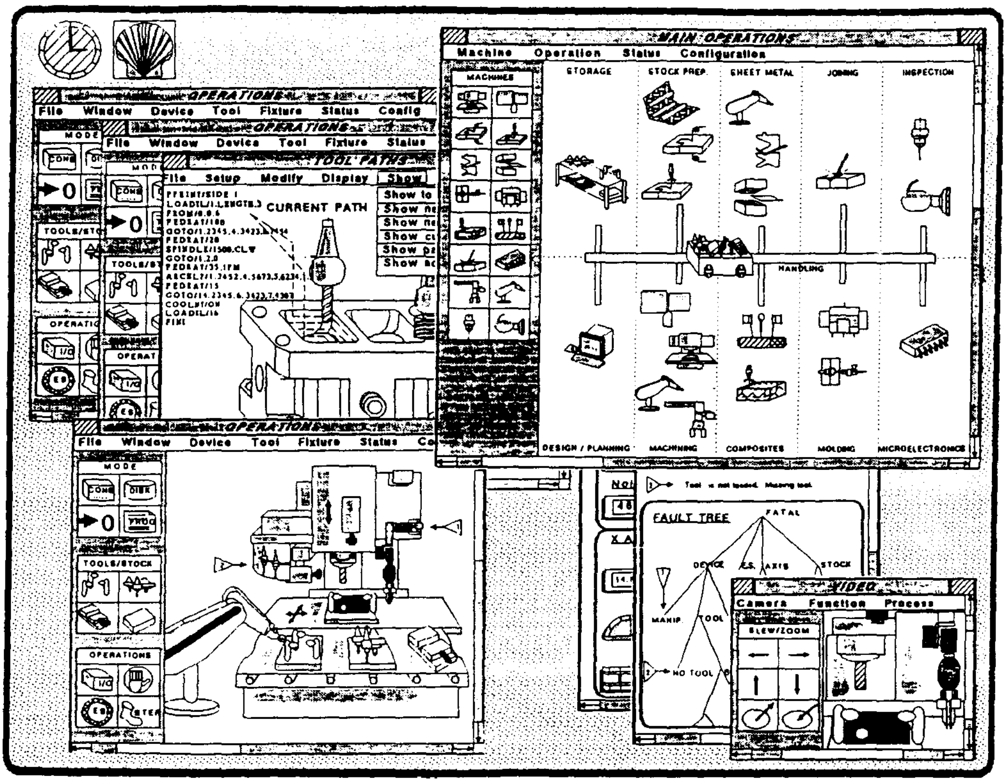

Figure 6. The User Interface - The Plant Operator Screen

In an oversimplified way, some aspects of factory life are best "seen" with left brain hemisphere skills [17], where digital-like analysis is performed; other aspects are best "seen" with right brain hemisphere skills, where visuo-spacial and analogical skills are performed. Of course the beauty of the

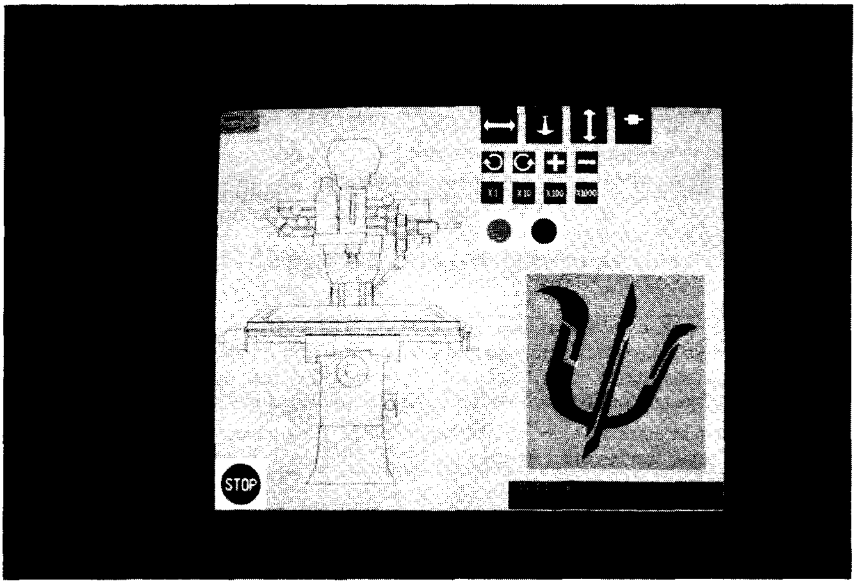

Figure 7. The Machine Operator Screen - the MOSAIC Project Implementation

human mind is that these different views can be integrated, especially with some years of experience in a factory, and used collectively to make global decisions. And even if different people are involved in the decision making, potentially "seeing" the factory in different ways, language and human communications are usually flexible enough to work through to a common understanding despite initial confusions and the inadequacies of human language. The integration of factories with computers, rather than people, however, brings a new focus on the above issues. Computers cannot so easily deal with ambiguity, double meanings and different ways of "seeing" the same functions. Even though expert systems have been created, say to plan machining functions [11], they are written in languages different from the APT programs needed to cut the part [6] and the robot programs [7] needed to reposition the part in a vise. Figure 8 highlights these differences for some processes. Realistically, since the application of expert systems to manufacturing is a relatively new endeavor, the knowledge bases are, today, currently much too small to relate to the true needs of the factory. Thus the "vertical integration" of functions as different as CAD, planning, real-time machining and inspection has not been possible in computer integrated manufacturing. Even the "horizontal integration" (interfacing at the flexible manufacturing cell level) of different machines from many different vendors has been a major challenge and only recently seen industrial successes [9].

The above discussion represents a ***needs analysis*** for a high level manufacturing language. This language should not only integrate different levels of the factory but also allow an environment such as MOSAIC to be generically programmed. It should be a meta-language that can describe, with equal ease, CAD methods, machining functions, robot functions, vise functions and sensor activities before some translation to a lower level (or device-level) language occurs.

An initial case-study of such a universal manufacturing language is now presented. The goal is to describe a meta-level object-oriented language which will serve as a *lingua franca* within the manufacturing domain, thereby rendering the programming requirements of individual devices mostly transparent to the software developer and user.

In the above discussion, it was emphasized that those aspects of manufacturing involving visuo-spacial skills are most easily "seen" with the right brain. An examination of common undergraduate

Issue	Machine-ing	Robot-ing	Vision-ing
functionality	milling, forming, injection...	handling, assembly...	guidance, monitoring, inspection...
part description	part production		part recognition
workspace description	part location	motion planning	scene recognition
motion planning	part-specific, preprogrammed	workspace-specific, continuous	
motion coordination	tightly interpolated	loosely interpolated	
speed	slow to medium	medium to fast	slow to medium
accuracy	medium to high	low to medium	low to high
sensing	limited to the process	highly diversified	
descrete operations	limited to the process	highly diversified	
language features	part-oriented, static-environment	workspace-oriented, dynamic-environment	both part and workspace oriented, dynamic-environment

Figure 8. Typical Features of Different Manufacturing Processes (machine-ing, robot-ing and vision-ing)

texts on 'manufacturing processes' reveals many iconic images of diverse processes, such as machining, welding, powder metallurgy, plastics extrusion and composite manufacturing [18,19]. These processes are first described in global terms, with "pictures". It is only the finer details that can be described more quantitatively with equations of the form found in stress analysis and heat transfer analysis texts.

Figure 9 is a first attempt to view all the "pictures" in these common undergraduate texts on manufacturing in the same framework. At the same time, the goal is to provide an abstraction of any

Machine	Tool	Process	Pre-Process Object		Post-Process Object
Nominative	Ablative	Verb 1	Accusative 1	Verb 2	Accusative 2
John	uses a knife	and cuts up	a lettuce	to make	a salad.
A milling machine	uses a tool	and cuts up	a block	to make	a part.
A welding machine	uses a torch	to weld	two plates	to make	a bracket.
A powder-metal cell	uses a die and furnace	to sinter	powder	to make	a composite.
A forming machine	uses a die	to bend	sheet	to make	a panel.
A rolling mill	uses rolls	to reduce	a plate	to make	a ship-plate.
An extrusion press	uses a die	to extrude	a billet	to make	an extrusion.
A plating bath	uses electrolyte and power	to plate	a block	to make	a plated part.
A heat-treatment furnace	uses gas and power	to carburize	a block	to make	a hardened part.
A robot	uses a gripper	to assemble	parts	to make	an assembly.

Figure 9. The Syntactical Structure of a Human-like Manufacturing Language

Machine	Tool	Process	Pre-Process Object		Post-Process Object
The CAD system	uses cad algorithms	to process	data	to make	a tool path.
The expert system	uses rules and databases	to process	knowledge	to make	a setup plan.
The loading robot	uses a gripper	to position	cutting tools	to create	a tool-set in the carousel.
The loading robot	uses a gripper	to position	the vise	to create	the generic set-up.
The loading robot	uses a gripper	to position	the stock in the vise	to create	the first set-up.
The vise	uses automatic jaws	to clamp	the stock	to create	a fixed setup.
The milling machine	uses tool 1	to cut CNC routine 1	into the block	to make	feature 1.
The milling machine	uses tool *n*	to cut CNC routine *n*	into the block	to make	feature *n*.
The dextrous manipulator	uses a gripper	to reposition	the stock	to create	new setups *n* to *n* + 1
The vision system	uses algorithm 1 and database 1	to analyze	the vise area	to verify	a chip-free setup environment.
The vision system	uses algorithm 2 and database 2	to analyze	the flank of the tool	to verify	tool conditions.
The piezoelectric dynamometer	uses algorithm 3 and database 3	to analyze	tool forces	to verify	tool conditions.

Figure 10. Application of the Manufacturing Language Structure to the MOSAIC Environment

generic manufacturing process in such a way as to be a basis for the universal manufacturing language. The icons of the machining process are the best place, from a manufacturing engineers' perspective, to begin analyzing Figure 9. A generic machine tool (A) holds a tool of some geometric description (B) which is then set into motion (at a predescribed speed [v], feed [f] and depth-of-cut [d]) to cut a raw block (C). A combination of such cuts leads to the formed part (D).

A *sentence* describing these activities is placed above the icons. And below the icons many other common processes are described in the same way. At the bottom of the table, we point out that plating, heat-treatment and assembly can be viewed with the same paradigm.

In the most abstract form, the top line of the table shows that a *machine*, by way of a *tool*, applies a *process* to a *pre-process object* to *make* a *post-process object*. In language structure, these terms correspond to (in the second line of the table) the nominative, ablative, verb 1, accusative 1, verb 2 and accusative 2. The sentence, 'John uses a knife and cuts up a lettuce to make a salad' is also shown for comparison.

Figure 10 extends such ideas to the more specific environment of MOSAIC. Each device can be treated in the same way as the machine tool. However a 'device' can also be viewed more abstractly. Hence the expert system is a device that uses rules and databases to process knowledge to make a setup plan. Going down the table, a somewhat chronological series of events is described involving the various functions of the loading robot, the vise, the milling machine and the dextrous manipulator. Note that the vision system and other sensors can be treated similarly.

The table and its extensions is the structure for the MOSAIC programmer. The programmer can use the *names (objects)* in the table and arrange them in sentences to program *actions*. Sentence construction is done by 'clicking-on' named devices in the windows shown in Figures 4-6 and specifying their actions from 'pull-down' menus. Timing and program flow-structure provisions, needed for real-time operation and for decision making, should be incorporated by extending the syntactical structure to support logic.

5. CONCLUDING REMARKS

Next generation machine-tools and their controllers will have to cope with a highly diversified and dynamic device and sensor environment. The paper has briefly presented the controller configuration being currently implemented in the MOSAIC project and on two major software issues, namely the user interface and the manufacturing language. These represent our conception of the future man-machine interaction, using visual and human-language tools.

We plan to continue integrating the various data, knowledge and control components in the open-system controller, and support it with prototypes of the manufacturing language and the graphic interface. Future research will include the understanding of the human-like aspects of the language, and its implementation via an existing real-time high level computer language.

6. ACKNOWLEDGEMENTS

The research described is part of a broad program. To support laboratories, equipment and personnel we acknowledge the following organizations: The National Science Foundation, Grants DCR-8320085, and DMC-8611150; The Office of Naval Research, Grant N00014-82-K0381; New York University; and The Institute for Defense Analyses (IDA).

We thank Patricia Bedard and Isidore Rigoutsos for the implementation of the graphic interface, and Philip Strongin for the discussions on the manufacturing language.

7. REFERENCES

1. I. Greenfeld, F.B. Hansen and P.K. Wright, *Self-Sustaining Open-System Machine Tools,* NYU - Courant Institute of Mathematical Sciences, NYC, NY (September 1988). to be presented at NAMRC XVII

2. I. Greenfeld, *MOSAIC - A Machine-tool's Open-System, Advanced, Intelligent Controller,* NYU - Courant Institute of Mathematical Sciences, New York, NY (April 1986). to be submitted to the ASME Symposium on Control Issues in Manufacturing Processes

3. U.S. Airforce, *Next Generation Machine-Workstation Controller,* Materials Laboratory, Air Force Wright Aeronautical Laboratories, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio, 45433-6533 (November 1987).

4. Cincinnati Milacron Marketing Company, *Intelligent Machining Workstation Initiative (IMW),* Materials Laboratory, Air Force Wright Aeronautical Laboratories, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio, 45433-6533 (1988). Technical Reports prepared under Air Force Contract #F33615-86-C-5038

5. J.U. Korein et al., "A configurable System for Automation Programming and Control," *Proceedings of IEEE Conference on Robotics and Automation,* IEEE, (April 1986).

6. I.H. Kral, *Numerical Control Programming in APT,* Prentice-Hall, Englewood Cliffs, NJ (1986).

7. L.R. Nackman, et. al., *AML/X: A Programming Language for Design and Manufacturing,* IBM Thomas J. Watson Research Center, Yorktown Heights, NY (July 1986). Research Report 11992 (#54006)

8. T. Lozano-Perez, "A Language for Automatic Mechanical Assembly," *in Artificial Intelligence: An MIT Perspective,* pp. 245-271 MIT Press, (1979).

9. D.A. Bourne, "CML: A Metta Interpretor for Manufacturing," *AI Magazine* 7 No.4 pp. 86-96 (1986).

10. D.G. Johnson, *Programmable Controllers for Factory Automation,* Marcel Dekker, Inc., NYC, NY (1987).

11. C. Hayes and P.K. Wright, "Automated Planning in the Machining Domain," *Knowledge Based Expert Systems for Manufacturing* 24 pp. 221-232 American Society of Mechanical Engineers, (1986).

12. P. Robinson, "Power to the Process," *Computer Graphics World* **vol. 12, No. 3** PennWell, (March 1989).

13. National Instruments, *Data Acquisition and Analysis Using Next Generation Personal Computers,* National Instruments, Austin, Texas 78727 (1988). Seminar Manual

14. A. Goldberg and D. Robson, *Smalltalk-80: The Language and Its Implementation,* Addison-Wesley, Reading, MA (1983).

15. U.S. Department of Defense, *Ada Reference Manual.* July 1980.

16. B. Stroustrup, *The C++ Programming Language,* Addison-Wesley, Reading, MA (1986).

17. D.G. Bobrow, *Qualitative Reasoning about Physical Systems,* MIT Press, Cambridge, MA (1985).

18. S. Kalpakjian, *Manufacturing Process for Engineering Materials,* Addison-Wesley, Reading, MA (1984).

19. E.P. DeGarmo, J.T. Black and R.A. Kohser, *Materials and Processes in Manufacturing,* Macmillan, New York, NY (1984).