OPEN-SYSTEM MACHINE CONTROLLERS – THE MOSAIC CONCEPT AND IMPLEMENTATION

I. Greenfeld
Courant Institute of Mathematical Sciences
New York University
New York, New York

ABSTRACT

A generic concept of an open-system controller for machine-tools is described. The concept is proposed as a major factor in the revitalization of the U.S. machine-tool industry. A multi-device multi-sensor environment of future machine-tools is projected as the main technical need for a multi-functional flexible controller. The controller is conceived as a wide-scope system, responsible for highly automated design, planning and production. The architecture of the controller is based on general purpose computers and computer equipment, rather than on proprietary equipment, thus keeping the machine industry in pace with the rapid changes in the computer industry. Design goals for the controller's development are suggested, mainly the integration of a fully equipped system, the development of a fast real-time operating system suitable for manufacturing, and the development of an advanced manufacturing language for handling intelligently a diversified environment. The first phase of the NYU implementation of the project is described. A Sun workstation, expanded with additional real-time processors, motion controllers and various devices, has been integrated with a milling machine environment. The concept, though, is extendable to other complex machine environments. Future enhancements to the system include control of manipulators and programmable fixtures, and a real-time vision system.

1. INTRODUCTION

This paper describes and discusses ideas and concepts concerning advanced controllers for future machine tools. In the center of focus stands the **Machine-tool Open-System Advanced Intelligent Controller** (MOSAIC), which is a generic-type controller concept, based on an open-architecture computer system, and intended for use on machines that are highly self-sustaining. The controller is conceived as a wide-scope system, responsible for highly automated design, planning and production. The MOSAIC concept is currently being implemented at NYU on a milling machine environment, and named the MOSAIC Project. The concept, though, is extendable to any complex machine environment.

The introduction of open-system controllers is suggested as a key technological factor in the revitalization of the U.S. machine tool industry. The competitive status of this industry has been investigated by the National Academy of Engineering [1]. It is shown that the U.S. manufacturers have lost a considerable fraction of the standard machine-tools' market to foreign competitors, in particular to the Japanese. Although sales figures show a recent increase in quantity, still, approximately 50% of machine tool sales in the U.S. are from offshore suppliers [2]. The foremost contributors to the success of foreign companies are the controllers installed on their machines. These controllers, most notably Fanuc's (which is installed also on the major part of U.S. made machines), are superior to their american counterparts in performance, functionality and reliability. Although in mechanical design there is no foreign advantage, the controller is the main constituent of the machine's performance, and therefore a central consideration in a machine purchasing decision. It could almost be said: *The Controller is the Machine.*

An analysis of the historical, economical and technical background [3], suggests that the industry has not benefitted enough from the rapid technological advances in computer hardware and software, due to incompatible cultures and needs of computer companies and machine-tool companies. This gap may be bridged by implementing machine controllers with general purpose computers and computer equipment, thus opening a new and vital market for these computers, and possibly urging both industries to collaborate through their mutual needs.

Market forces are ever-dictating lower-price higher-quality higher-diversity products. In terms of manufacturing, this means small-batch manufacturing processes with extensive automation and in-line quality control. Therefore, future machine tools and factories will present a totally different environment to the controller. In particular, it seems that machine tools are going to be more *self-sustaining* and *open-system*, meaning that they will have a considerable amount of hardware and software enabling unattended operation, and, a standard and flexible configuration. These features are discussed in detail in [4] and in section 2.

The extent to which a certain manufacturing process should be unattended (and consequently, a machine - self-sustaining) is subject to economical considerations. While extensive automation of regular events, such as process design, process planning and preprogrammed (though somewhat adaptive) production, is plausible, it may still be more efficient to man unexpected events associated with complicated error-recovery procedures. The slowness in which FMS's (Flexible Manufacturing Systems) have been accepted by the market, may be accounted for by the immense integration difficulties posed by a multi-vendor, disparate equipment environment. This is where open-system equipment can make a difference.

Open systems are usually expected to have at least the following features:

- **connectivity** - the ability to cooperate with other computers over a network .
- **availability** - wide distribution of executables, source code, hardware, and documentation.

• **expandability** - modularity and scalability of hardware and software.

• **portability** - the ability to readily install the software on different computers.

The degree to which a system should be *open* is somewhat a matter of a trade-off between high performance and a universal interface. This question of ballance is especially acute in the machine-tool industry, where high performance (e.g. speed, accuracy, reliability) is essential, while the market is comparatively small and unable to support development of a full line of specialized products. The foremost example of open systems is the PC and workstation computer industry, with companies like IBM and Sun Microsystems. Sun has introduced a line of workstations, based on the Motorola 680*0 microprocessor family, which support standards like the C programming language, the Unix operating system, the VME data communication bus, and the Ethernet local area network. Through these standards, and other advanced features like graphic tools and network computing, these workstations from Sun and other companies, have become a success story, maintaining U.S. leadership in the world of computers. Due to the open-system architecture, a large number of third-party hardware and software companies have flourished, spreading the basic structure of the industry widely, creating positive competition, with the outcome of a large variety of high performance-to-price products.

The MOSAIC controller (introduced in [5]) presented herein is specified and designed especially for the projected self-sustaining open-system future environment of machine tools. The concept (and it NYU implementation - in brackets) is based on a general purpose computer (a Sun workstation), running a standard operating system (Unix), programmed in a standard programming language (C), and having a standard hardware (VMEbus). It provides a flexible environment that can drive the machine, the manipulators and the sensors. It can be configured as a controller for a single machine or for multiple machines. It uses the most advanced features of today's computer technology, such as high and expandable computational power, graphic user interface, and support of a variety of communication networks. It benefits from the huge market of hardware and software of standard computer equipment provided by third party vendors. Also, with the current trends in the general purpose computer industry, the controller is expected to have a good price-performance ratio.

Other major machine controller development programs exist, most notably the U.S. Air Force Next Generation Machine-Workstation Controller (NGC) [6] and the Intelligent Machining Workstation (IMW) [7]. The NGC program entails hardware and software design over the span of 6 years and $100 million. It covers a wide scope of controller functions and architecture, and proposes a development and implementation plan. Recently, an RFP (Request For Proposal) has been issued. Also, the National Center for Manufacturing Sciences (NCMS), a consortium of about 100 companies supporting research in manufacturing technologies, has been working with the U.S. Air Force on the NGC. The program emphasizes the open-system design, and suggests that the controller will be configured as a modular system, the components of which will have a standard interface, and will be manufactured by the various cooperating companies [2]. It is not clear whether the program will stimulate the participation of the computer industry, or which computer standards will be adopted. The consortium solution seems not be the best way to efficiently exploit the capabilities embedded in the wide structure of the computer industry. Hopefully, concepts like MOSAIC, which rely heavily on the computer industry, will contribute to this important program.

The IMW program is concerned predominantly with intelligent software development, assuming the use of mainstream current controller hardware design, along with projected controller improvements such as a MAP interface to allow dual-way communications. This program is also supported by the U.S. Air Force, and is carried out as a collaboration between Cincinnati Milacron company, Pratt & Whitney company and Carnegie-Mellon University. The program includes the development of a family of expert systems for planning and cutting (planner, cutter, holder, sensing), languages for part description (MFGMAP) and feature exchange (FEL) in conjunction with a solid modeler, and a supervising control and human interface. Some uncertainty as to the hardware configuration exists, which slows down the design and implementation of the control scheme. Most of the software developed under this program is expected to be portable, and may be used in projects like MOSAIC and others.

Another research program is IBM's Configurable System for Automation Programming and Control [8], which adopts the open-system approach, but still relies on the development of special proprietary communications hardware.

Section 2 of the paper describes in more detail the future diversified environment of machine tools, and the shortcomings of current controllers. Section 3 and 4 contain the conceptual layout of the open-system controller and the design goals involved with its implementation. And, section 5 brings some details of the NYU MOSAIC implementation.

2. THE FUTURE ENVIRONMENT OF MACHINE TOOLS

An interesting view of **self-sustaining open-system** machines is presented in the book Manufacturing Intelligence [9]. The machine is decomposed into human-like functions: the manufacturing ***hand***, the manufacturing ***eye***, the manufacturing ***brain***, etc. A technically oriented viewpoint is shown in figure 1.

A self-sustaining machine tool is equipped with dextrous manipulators dedicated for the continuous needs of machining, such as chip clearance and part relocation. A variety of sensors provides vision, touch, force, noise and temperature senses, with the tasks of recognizing events, in-cycle inspection and optimizing the machining parameters. A rich supporting design environment is essential, including:- cadcam for part and tool path design; expert systems and libraries of technical information for an optimal and efficient design; and an operating environment with process planning capability.

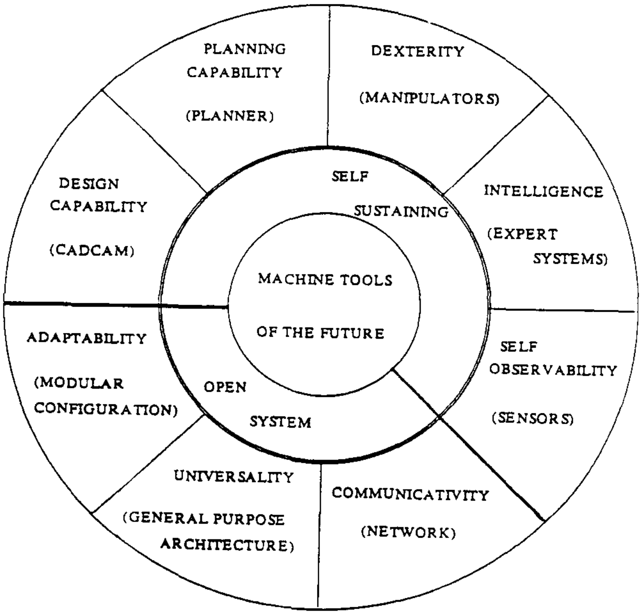

Figure 1. Next Generation Machine Tools - Open-System and Self-Sustaining

As an open-system, a machine tool is equipped with a general purpose computer, such as a PC or a workstation, enhanced with additional standard computer hardware, that controls the axes of motion and the various devices, and manages programs and data. Communication utilizes networks that are universally accepted in the computer world, and thus resources may be shared as needed. The machine is adaptable to the changing environment and tasks, in terms of its controller's computer configuration, and in terms of the mechanical construction.

Refer to figure 2 for a comparison of the current machine controller environment to the next generation machine environment. Figure 3 shows an example of devices mounted on a milling machine (from the MOSAIC project). The difference between current controllers and next generation controllers is mainly in scope. While current controllers are dedicated predominantly to low level control of motion and tools, and rely on interfaces with higher level factory systems, next generation controllers are conceived as full-scale systems for design, planning, production and ispection, as well as for low level control and communications. It is the difference between *horizontal integration* (**interfacing** between disparate systems) and *vertical integration* (**integrating** hierarchical functions into a unified environment). In that sense, the traditional meaning of the term *controller* is insufficient, and merits refreshing.

Although a full-scale integration has not been implemented yet, many of the individual technologies and components of such unattended machine tools already exist as lab experimental devices or as commercial products. A cursory list follows: the DexMan dextrous manipulator [4]; various sensors for machining [10]; the FLECS automated fixturing system [11]; the Valysis SoftGauge for in-cycle gauging [12]; adaptive control; advanced cadcam applications [13]; the Machinist expert system for setup planning [14]; the CarrLane library of tools and fixtures [15]; various manufacturing languages such as AML/X [16] and CML [17]; general purpose computer technology by companies such as IBM and Sun; the MAP/TOP protocol [18] and various other communications networks for manufacturing [19];

The challenge in the design of the controller is in providing the appropriate computer environment for the integration and implementation of a complex machine tool environment that applies these individual technologies. This challenge cannot be met with current controller technology.

While today's Computer Numerical Controllers (CNC's) are adequate for servo control of motion axes and control of specific descrete devices, they are very limited in terms of programming flexibility and in terms of communications with external computers and devices. Standard configurations cannot accommodate non-machining devices such as workholding accessories, force sensors, vision sensors, and other subsidiary devices. Although using advanced electronics, CNC's design concepts are conservative, especially in terms of hardware interfaces and user interfaces. Computer devices and architectures such as the magnetic disk storage and data buses have only recently appeared (even then, usually as proprietary products), and computer innovations such as the latest microprocessors are always late to appear. Current CNC communications are mainly through slow serial lines (RS232 and similar ones), reading and sending files, without the capability for real time control. Support of advanced communication networks (MAP) is expected to be introduced only during 1989, and even then the effectiveness of the communications will depend on the openness of the controller's software towards the host computer.

The user interface in today's controllers consists of interacting with a push-button, upright operator's panel (the new touch-sensitive screens are essentially the same). Programming is through a cryptic controller-specific G-code language, which is not practical for non-trivial programs, and necessitates a special, installation dependent, post-processor to translate from higher level part and tool programming languages such as APT.

3. THE OPEN-SYSTEM CONTROLLER

Figure 4 presents a generic architecture of the controller. A multi-machine configuration is shown, which consists of a *Plant Controller*, a few *Machine Controllers*, and machines. A *Local Area Network (LAN)* establishes communication between the plant controller and the machine controllers, and with other plant controllers in the factory. The machine controllers are directly connected with the machines and the devices and sensors mounted on them. They execute the real-time control operations such as machine-tool axes' motion, manipulators' motion, part clamping, sensor data acquisition, and vision. The plant controller is dedicated for the supervision of a few machine controllers, serves as an operator's terminal, and provides file systems and local and wide area communication services. It also serves as a design and programming workstation.

Both controllers use an industry standard data bus such as the VMEbus, the PCbus or the Multibus. They communicate over a LAN such as MAP/TOP and EtherNet (both typically yield appx. 10 Mbits/sec) or the expected FDDI fiber-optic LAN (estimated 100 Mbits/sec). For single machine configurations, a direct connection of

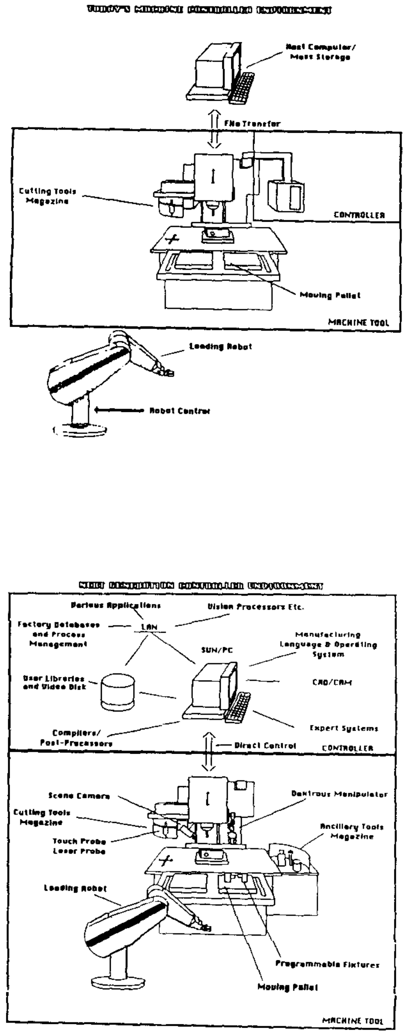

Figure 2. A Comparison: Today's Machine Controller Environment and Next Generation Controller Environment

the two buses may be obtained through a bus repeater, thus providing a very fast data transfer rate (yields appx. 300 Mbits/sec). All these data transfer rates are theoretical maximum values. Practical speeds may be up to four times slower, depending on the physical configuration and the nature of the line's traffic. Usually, direct bus connections, which limit the distance between the two controllers to a few feet, will not be necessary, since the real-time work is done mainly on the machine controller. Exceptions to that are development systems, such as the current MOSAIC implementation (see section 5), where an extended bus provides higher flexibility. The design of the system may be such that this variation in the bus configuration is transparent to the user.

Communications on both sections of the bus are supported by communication processors that relieve the load from the real-time control processors, by performing file transfer, data compression, command interpreting, etc. The real-time computations are performed by processors running a real-time operating system, and executing the user application programs. Current processors are based on microprocessor technology such as the 68020. Specific operations such as motion cotrol, image processing, and various I/O operations are performed by dedicated processing systems. Workstation services, like file systems and terminals, are provided by a general purpose computer such as a Sun or a PC, supporting the variety of compilers, appli-

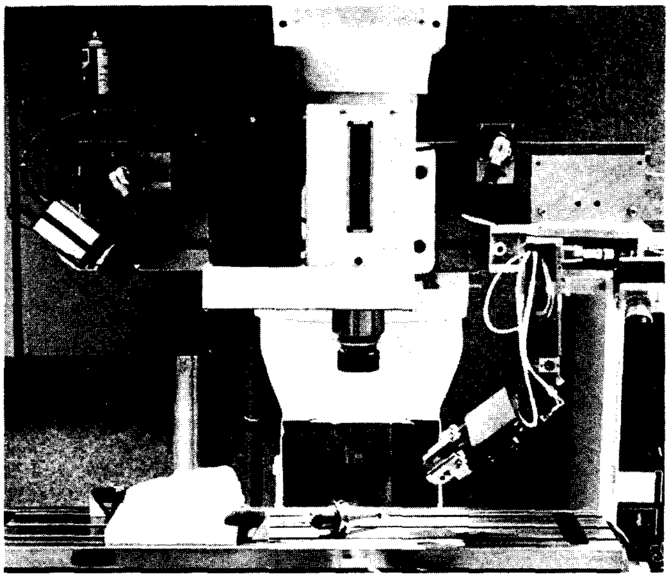

Figure 3. Devices and Sensors on a Machine Tool - a Dextrous Manipulator (DexMan), Cameras, and a Touch Probe (not shown) - the MOSAIC Project

cation programs and utilities available on the market. Some of the services may also be provided by an auxiliary workstation hooked to the machine controller, located at a site close to the machine.

All the systems and sub-systems mentioned are available **today** on the market from multi vendors, and de facto industry standards are emerging. This is not the case with the specialized software, mainly the real-time operating system and the advanced manufacturing language needed to program and run the controller. These will eventually evolve along with the development of the open-system controllers as discussed in section 4 ahead.

4. DESIGN GOALS

The specification and design of the controller involves a variety of hardware and software technologies. The research is intended to investigate the broad implications of such a controller on machining languages and operating systems, as well as on integration with machine tools and devices. Currently, three main research and development areas associated with this open-architecture system have been identified:-

i) The integration of the *Open-System Machine Tool Controller*. It includes the integration of a general purpose computer and a machine tool with new devices and sensors, such as those mentioned in this paper. These devices are part of the machine-specific configuration, and are controlled locally by the open-system controller. In general, no special computer hardware has to be developed, since only general purpose equipment is used. Some improvements to specific equipment will be needed, though, for better performance and reliability. It is expected that with the future introduction of more powerful processors and workstations, and the development and commercialization of parallel and distributed-processing computer systems (see example in [20]) those architectures may gradually substitute elements of the proposed architecture. The ***openness***, though, is kept. More on the NYU MOSAIC integration is described in section 5.

ii) The development of a *Real Time Operating System for Manufacturing*, suitable for the very high speed control required for machining and manipulation operations. A real-time operating system coordinates multiple sensors, actuators and subsystems, communicates with other computers for high level tasks, and provides support for fuctions such as task monitoring and error-recovery. It should interface to industry standard operating systems such as Unix or OS/2, which provide high level management, filesystem operations, communications, and a good programming environment. Presently, although quite a few real-time operating systems exist, none is universally accepted. Also, the current general purpose real time operating systems do not provide the response time essential for maintaining the speed, accuracy and safety features inherent to machine tools. Typically, an update rate of all degrees of freedom is expected to be around one milisecond, thus allowing for high speed machining and manipulation. On the other hand, the exact time demands from the real-time operating system vary as a function of the hardware. E.g., as more powerful motion controllers are introduced to the market, the system integrator will not be confronted with the tough deadlines of motion control, but rather with the lesser demands (by an order of magnitude appx.) of command interpretation and process decisions. Thus, the specifications of the operating system will depend on the hardware. A few existing operating systems may be considered, among them SAGE (NYU) [21], a real time supervisory operating system for robot control, NRTX (Bell), Condor (MIT) and Sparta (IBM) [22]. Guidelines for the selection of an appropriate system are contained in [23].

iii) The development of an *Advanced Manufacturing Language* for manufacturing programming, that resides in the open-system controller. Some of the existing languages are APT (Automatically

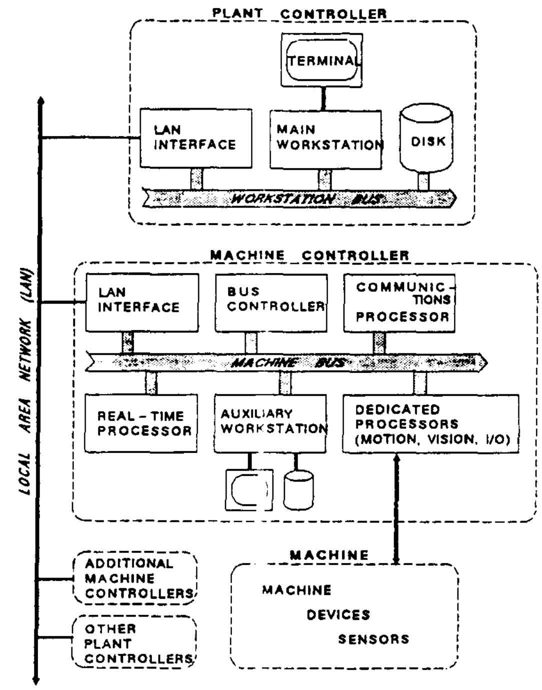

Figure 4. A Proposed Configuration for an Open-System Machine Controller

Programmed Tool) for machining [24], ladder logic for programmable controllers [25], AML (A Manufacturing Language) [16] for manipulating, and CML (Cell Management Language) [17] for coordinating various cell operations. While these should be supported, a more universal and flexible language is needed. It includes provisions for real time control needed for the operation of accessory devices in conjunction with the machining process and other manufacturing processes, a more direct connection to cadcam systems, and a flexible interface for user applications. Artificial intelligence techniques are to be used for the generation of process plans, and for adaptive control. The MFGMAP and FEL languages, now being developed under the IMW project [7], cover a similar scope, and may be adopted. At the user level interface, the manufacturing language will use a human language syntax and graphic tools. The user interacts with objects, an object being a device or an operation, represented graphically on the screen through icons and menus. The device-level implementation may still be using existing application languages like APT and AML, with enhancements for real-time coordination. Alternatively, the whole language software system may be implemented with a general purpose object-oriented real-time language like ADA, C++ or Smalltalk-80.

5. MOSAIC CONFIGURATION - NYU IMPLEMENTATION

This section describes the implementation of the MOSAIC concept in the Robotics and Manufacturing Laboratory at New York University. An enhanced milling machine environment was created for the purpose of researching and demonstrating the concept in operation, under various demands of performance. The concept, though, is applicable to other complex manufacturing environments.

Note: It should be emphasized here, that for the sake of providing a full picture of the project, both the current **and near future** hardware and software configuration (figures 5 and 8) of the project are shown. Refer to details on the current status in the following text.

5.1. Hardware

Figure 5 shows an overview of the MOSAIC system configuration. It describes the complete system, which is currently in its first phase of construction. This configuration is slightly different from the generic configuration presented in figure 4. The plant controller is termed *supervisory controller* due to the single-machine configuration. The VMEbus is repeated directly rather than through a LAN, but Ethernet is supported just the same by the real-time system.

The milling machine, a SP-1 from Intermark Hartford, has 3 travel axes and a spindle, all using AC induction motors and encoders from General Motion. The system uses a Sun 3/160 workstation with a 19" graphic monitor. The Sun's 9U VMEbus is extended to a standard 6U VMEbus through the VMEbus Repeater 2000 from HVE. The Sun provides file system and communication services, including a connection to the lab's Ethernet. This enables the access to the vast variety of other computer services available in the lab, including technical and programming libraries, the Anvil-5000 Cadcam system, a Vicom image processor, a WORM video-disk, and more.

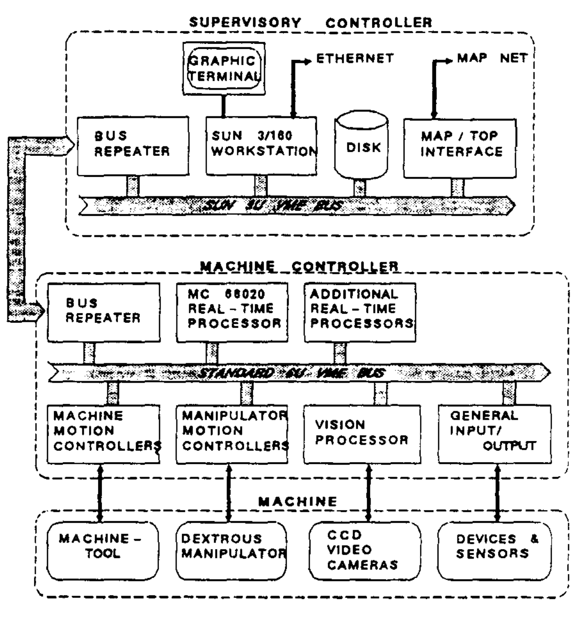

Figure 5. An Overview of the MOSAIC System Configuration - Current and Planned

The extended VMEbus rack, from Bicc-Vero, hosts a Pacific-Microcomputer 68020 CPU board with a 68881 FPU, which runs the real-time operating system and the application programs. Dual-axis VME-EG0 Motion Controllers from Creonics are used for controlling the machine's 3 axes and spindle, all with full position and velocity control, using a PIVF (Proportional, Integral, Velocity Feedback and Velocity Feedforward) digital control scheme. Additional processors and motion controllers are planned for the operation of the Dexman manipulator (shown in figure 3). A real-time image processor board is planned for monitoring, inspection and manipulator guidance. The video data will also be projected onto a window on the Sun's screen. Programmable fixtures and various sensors and devices are also planned.

Figure 6 shows the current equipment layout in the lab. The computer system is remote from the machine itself, and the computer terminal - monitor and keyboard - serves as the machine operator's panel. Video cameras provide a continuous monitoring of the cutting scene. All the system's components were purchased off-vendors'-shelf. The electrical connections were all designed and done in-house to retain flexibility in integration. Figure 7 presents photos of the overall environment and of the computer equipment.

5.2. Software

Figure 8 shows the MOSAIC software structure. Five categories are specified, including, from the top level to the bottom level: a graphical user interface; languages; planning applications; control applications; and machine applications. The low level **machine applications** are mainly vendors' proprietary software running on dedicated motion controllers, vision processors, etc. The **control applications** software provides motion control, from the single-axis control, to linear, circular and spline interpolations (multi-axis coordinated motion). It also provides control of the analog and descrete devices and sensors, like programmable fixtures, force sensors, vision, etc. The **planning applications** include motion planning for cutting complete surfaces. This feature is currently available only as an off-line tool-path generation by the Anvil-5000 Cadcam system. Future research, involving an expert system for machining (Machinist [14]) and computational geometry techniques, may provide a process level of motion planning. We plan also to integrate the expert system and the Cadcam system with the controller to provide quasi-real-time adaptive control of operations. A closed loop force control is now being implemented by measuring motor torques indirectly (in current mode) and varying the travel feedrate and spindle speed on the fly.

On the **languages** level, we currently have interpreters for APT and for AML. These are done in real-time, eliminating the need for the traditional post-processing. We plan to concentrate efforts on the introduction of a universal manufacturing language to enable us to control the machine, the manipulator, the vision system, and the other sensors and devices, concurrently.

On the top level, we have a graphical **user interface**, consisting of three interaction screens: the Designer Screen, the Machine Operator Screen, and the Plant Operator Screen. Each screen uses a few application windows (operations, monitors, cadcam, planning, etc.). The graphics is Sun/Mac style, using windows, icons, dialogue boxes, and pull-down menus. Currently, as a first phase, we have parts of the operator's screen, implemented experimentally under an X-Window environment.

The higher level software is generally serviced by the Unix operating system running on the Sun. Most of the operating system design, and the application programs, were developed in C under Unix. The lower level software is supported by the Sage [21] real-time operating system mentioned before, which also supports the language interpreters. A Hierarchical Control System (HIC) [26], developed for robotic applications, provides an extensive library of C routines for real-time control of operations, starting from basic operations (e.g. single axis motion), up to object level operations (e.g. interpolations and motion planning).

6. CONCLUDING REMARKS

A window of opportunity now exists for the introduction of open-system controllers. The manufacturing industry is in need for universal controllers for FMS's, and general purpose computer technology is ripe for indutrial control applications. Slow evolution seems not to be sufficient - rather a breakthrough technology is required. The new machine controller architectures are essential for the future environment of manufacturing. Due to the complexity of computer

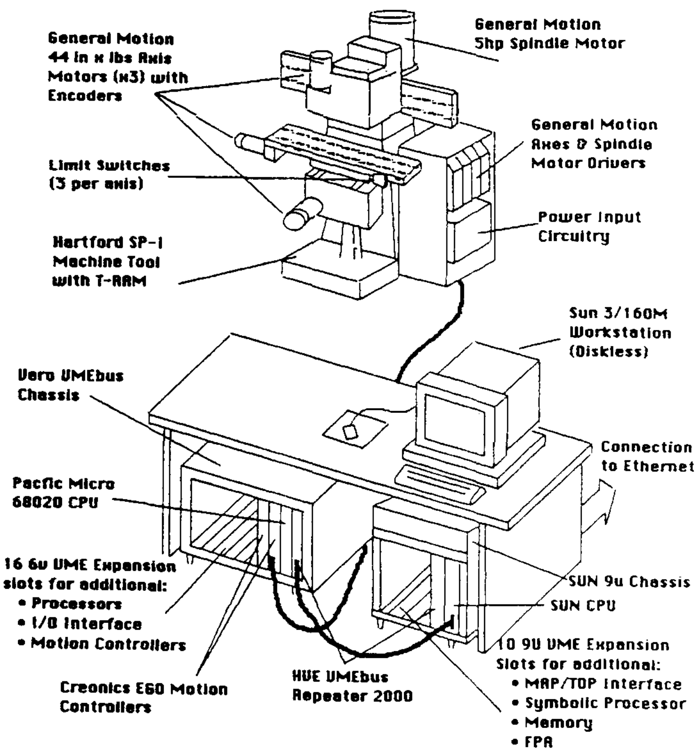

Figure 6. MOSAIC Current Equipment Layout

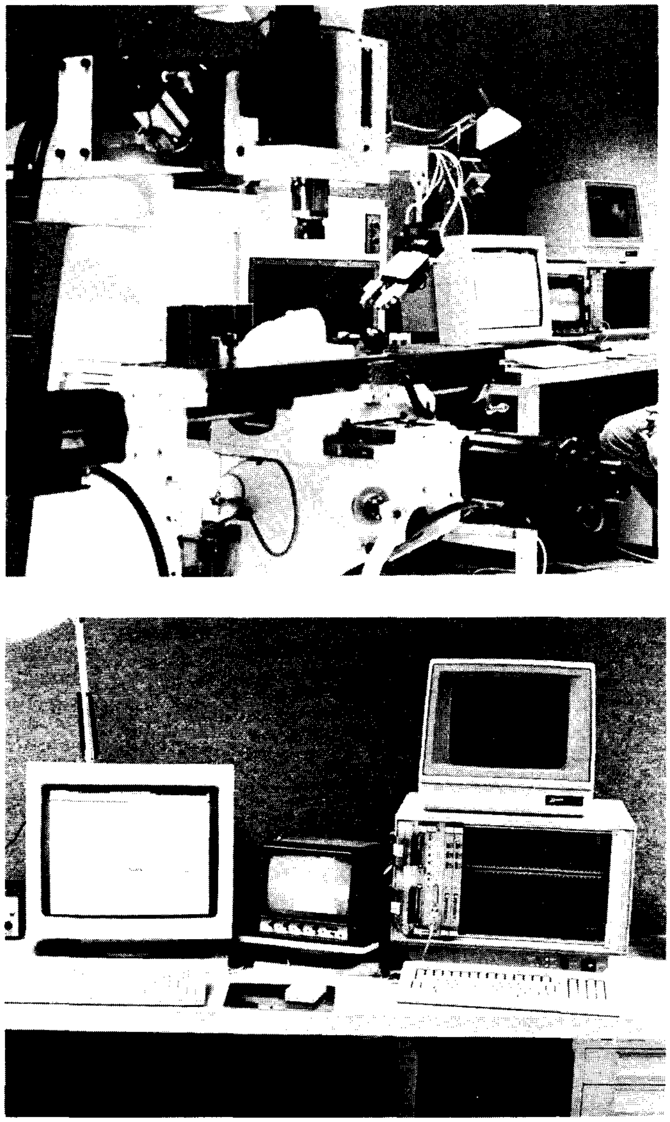

Figure 7. Photos of the MOSAIC Implementation: the Machine Tool and the Computer System

technology and its rapid pace of change, controller manufacturers cannot, any more, rely on their own resources to develop new and advanced controllers. The concept proposed here is centered around using standard computers and computer equipment as the backbone of open-system controllers. These controllers can support a variety of machines, devices and sensors concurrently, in a single-machine or a multi-machine configuration.

We plan to expand the NYU implementation presented in this paper by adding robotic devices, vision, and various sensors, and to demonstrate the flexibility of the system. We shall also direct efforts towards the improvement of the real-time operating system, and the introduction of a universal manufacturing language.

7. 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).

I thank Fred Hansen and Jim Fehlinger for hardware and software support, and Dayton Clark for his suggestions as to the system configuration. Also, I thank Jehuda Ish-Shalom for his encouragement.

8. REFERENCES

References

1. The National Academy of Engineering, *The Competitive Status of the US Machine Tool Industry,* Office of the Foreign Secretary, National Academy of Engineering, 2101 Constitution Avenue, N.W., Washington, DC 21418 (1983). National Academy Press

2. R.N. Stauffer and J.V. Owen, "Zeroing in on Machine Tool Quality," *Manufacturing Engineering*, Society of Manufacturing Engineers, (April 1989).

3. P.K. Wright and I. Greenfeld, *Growth Opportunities in the Machine Tool Industry,* NYU - Courant Institute of Mathematical Sciences, NYC, NY (1988). (to be published)

4. 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)

5. I. Greenfeld, *MOSAIC - A Machine-Tool Open-system Architecture*

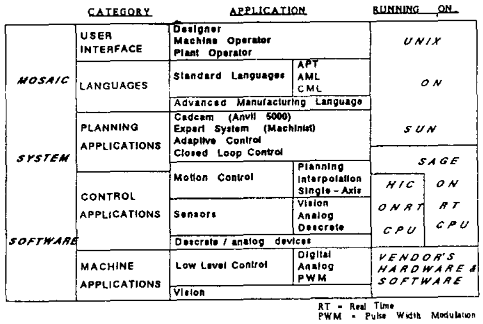

Figure 8. MOSAIC Software Structure - Current and Planned

Intelligent Controller, NYU - Courant Institute of Mathematical Sciences, NYC, NY (July 1988). Technical Note

6. U.S. Air Force, *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).
7. 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 and later reports). Technical Reports prepared under Air Force Contract #F33615-86-C-5038
8. J.U. Korein et al., "A configurable System for Automation Programming and Control," *Proceedings of IEEE Conference on Robotics and Automation,* IEEE, (April 1986).
9. P.K. Wright and D.A. Bourne, *Manufacturing Intelligence,* Addison-Wesely, Reading, MA (1988).
10. J. Tlusty and G.C. Andrews, "A Critical Review of Sensors for Unmanned Machining," *CIRP Proceedings* **32 No.2** pp. 563-575 (1983).
11. F.B. Hazen and P.K. Wright, "An Autonomous, Self-Loading Machine Tool," *Proceedings of the Manufacturing '88 Conference,* American Society of Mechanical Engineers, (1988).
12. Valysis Corp., *Softgauge etc.,* Valysis Corporoation, Santa Clara, CA (1988). (various brochures and newsletters)
13. W. Beeby and P. Collier, *New Directions Through Cad/Cam,* Society of Manufacturing Engineers, Dearborn, MI (1986).
14. 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).
15. CarrLane Mfg. Co., *Tooling Component Library in 3-D Iges Format,* CarrLane Mfg. Co., St. Louis, MO (1988).
16. 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)
17. D.A. Bourne, "CML: A Metta Interpretor for Manufacturing," *AI Magazine* **7 No.4** pp. 86-96 (1986).
18. World Federation of MAP Users Group, *MAP Reference Specification - Manufacturing Automation Protocol (based on GM MAP 2.2 Specification),* Society of Manufacturing Engineers, Dearborn, MI (June 1987).
19. K. Hughes and D.A. Dytewski, "Selection and Implementation of a Factory Automation LAN," *Autofact '87 Conference Proceedings,* SME, (November 1987).
20. J. Ish-Shalom, "A Broadcast Shared Memory for Robot Control Computations," *Computers in Engineering 1987* **1** The American Society of Mechanical Engineers, (1987).
21. L. Salkind, *The Sage Operating System,* NYU - Courant Institute of Mathematical Sciences, NYC, NY (August 1988). Technical report
22. J. Ish-Shalom and P. Kazanzides, "Sparta: Multiple Signal Processors for High Performance Robotic Control," *Proceedings of the 1988 IEEE International Conference on Robotics and Automation,* IEEE , (April 1988).
23. L. Salkind , *Unix for Real Time: Problems and solutions,* NYU - Courant Institute of Mathematical Sciences, NYC, NY (March 1988). Technical report
24. I.H. Kral, *Numerical Control Programming in APT,* Prentice-Hall, Englewood Cliffs, NJ (1986).
25. D.G. Johnson, *Programmable Controllers for Factory Automation,* Marcel Dekker, Inc., NYC, NY (1987).
26. Dayton Clark , *Hierarchical Control System* , Courante Institute of Mathematical Sciences , NY, NYC (February 1989). Technical Report No. 396