OSE: Open System Environment for Controller

OSE: Open System Environment for Controller

--- Development of an Open Architecture CNC with OSEC Specification ---

Satoru Fujita, Toshio Yoshida
OSE Consortium
Industrial Electronics & Systems Laboratory
Mitsubishi Electric Corporation

Summary

While networking and open systems have been progressing rapidly in PC and EWS-based information processing systems, significant barriers against this trend exist in the manufacturing sector, such as the processing of technical information and manufacturing know-how.

The purpose of the OSEC (Open System Environment for Controller) project is to build an architecture model, define control functions and interfaces in the model, make some inter-relation rules among them, and propose a new NC description language, validating it through prototypes. As a result of this project, it will be possible to provide capabilities and services in FA control equipment with open specifications among users, engineering companies, machine makers, controller makers and S/W companies. It will enable a multi-vendor business environment in the manufacturing sector and finally will lead to a future scenario of easy development / operation / maintenance and life cycle cost optimization for factories and end users.

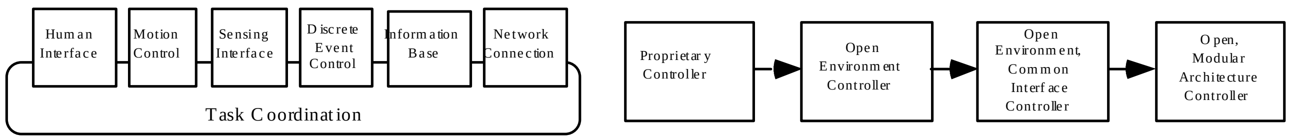
A major effort is being directed towards achieving this openness, with projects such as OSACA in the EU, and OMAC in the USA. The OSEC consortium consists of 17 companies including Japanese machine makers, controller makers, a computer company and one non profit organization for research. In this paper, the purpose and the contents of the OSEC-I & II projects since November, 1994 will be discussed.

1. Introduction

A lot of R&D is being done on open Computerized Numerical Controllers (CNCs) world wide, as a result of higher performance, down-sizing, lower costs, and the networking of computers due to the a rapid progress in semiconductor technology. Customer requirements are quite varied, including such diverse needs as integration of CAD / CAM through the use of networking, improvement of MMI and operatability, high level interpolation and intelligent machining with sensor fusion, and high-speed and high-accuracy machining. We expect that open CNC will be able to fulfill these requirements.

A major effort is being directed towards achieving this openness with projects such as OSACA in the EU, and OMAC in the USA. The OSEC discussed here is one of these projects. OMAC pointed out in [1] that open CNCs will have modular functions for manufacturing, and as a result, open CNCs will allow many high class vendors to join the manufacturing business, as well as let users build and reconfigure their own systems. In order to realize open CNC, architecture and standard interfaces between functional modules have to be studied. An international collaborative R&D effort is important in realizing these aims and in continually verifying ideas using current technologies.

This paper reports OSE's concept and OSEC-I and OSEC-II activities toward the achievement of an open systems environment and open CNCs.



(1) Functions of CNC **(2) Progression of OMAC**

Figure 1-1 Open modular, Architecture Controller

2. Why "Open CNCs" now ?

2.1 User Requirements

The CNCs in machine tools have been improving both in speed and accuracy by applying the advances in microprocessor technology since the 1970s. At the same time, these CNCs have been integrated with peripheral equipment such as MMI, ATC (Automatic Tool Changer), Instrument Equipment, and Palette Changer. It is a history of functional intensiveness for the machine tools. However because an open systems environment and a functional architecture have not yet been realized, the following user requirements have not been tackled, as described in [2] and [3].

Table 2-1 User Requirements

Items	Requirements
Reconfiguration	About 80 % of cylinder block machining in the auto industries doesn't need high accuracy and the only machining methods used are hole forming and face cutting. Although a user friendly MMI is necessary for various kinds of part machining in the job shop, the machining line doesn't need such MMI. However the machining line requires reduced manned labour and maximum automation. CNC functions should be added and deleted according to the usage conditions and the users
Scalability	The H/W and S/W of machine tools such as the control axis, special cycle programs, and the storage of NC programs need a functional independence and reconfigurability.
NC Language	The NC language format of EIA becomes very complicated in macro-programming and many special functions provided by NC makers. Since lots of problems occur in linking data with the CAD/CAM systems and the peripheral equipment, a new NC language should be provided.
New functions	CNCs need a curved surface interpolation in metal mold machining to avoid the need for the grinding process and need sensor feedback control for higher accuracy machining. A new business environment should be built for companies possessing high technology.
Intelligence	Tool exchanging is specified by the auxiliary codes but the makers of machine tools define different codes. Also the cutting conditions are different depending upon the rigidness of the machine tools. For compatibility of NC programs, CNCs should have intelligence to guarantee the same machining with other machines.
Openness	CNCs can not utilize 3-dimensional CAD/CAM and monitoring software. Manufacturing equipment should have the capability to integrate with design, operation, and sales domains, applying advanced technologies such as object orientation analysis and design, client-servers, and network communication.

2.2 Open CNC

Since we can assume that Personal Computer (PC) technologies will continue to lead the information processing domain toward open systems, we should build open system environments for manufacturing based on PC-related technologies. Network computing, interactive computing, object oriented design and analysis technologies would

be very important.
Manufacturing equipment such as machine tools demand high reliability and realtime capability. Therefore, Karatsu, Mitsubishi Electric and DELTA-TAU have brought a PC-based CNC to market, adding a NC board on the PC bus to guarantee higher reliability and realtime capability. The PC-based CNCs are more effective in S/W development and utilizing GUI functions and S/W packages of PCs, as well as more compatible with low-cost resources such as memory and disk space than conventional CNCs. The PC-based CNCs make it possible to link PC technologies to CNCs easily so we can regard them as a first generation of open CNCs. However PC-based CNCs still lack an open architecture and standard interfaces, and can not satisfy all user requirements. The most important things for the realization of open CNCs are reconfigurability of functions which the users require, portability of user systems to cope with changes due to the rapid progress of PC-related technologies, interconnectability for network computing and openness of the systems, and to build the business environment for multi-vendors, which is a prerequisite, by making architecture and interfaces of CNCs open.

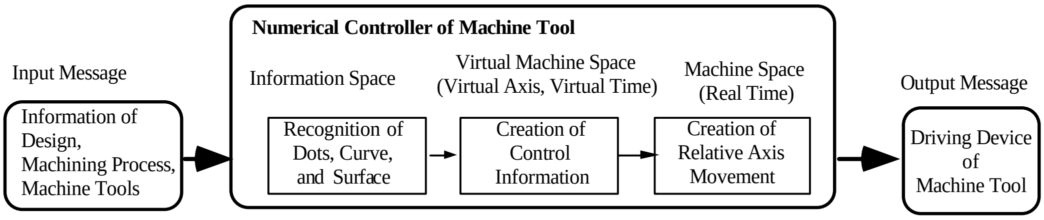


Figure 2-1 Fundamentals of CNCs

Prof. Wada of Setsunan University, a leader of OSE, says that "the fundamental part of CNCs is a kind of information conversion device which converts spatial geometry information as input messages to control information of the driving devices as the output messages", as shown in Figure 2.1 ([3]). The capabilities of machine tools have been advancing by integrating auxiliary functions for manufacturing around a basic function. However the conventional CNCs have become a blackbox, difficult to redesign. An objective of OSEC is to make future manufacturing systems open by undertaking R&D on the architecture and standard interfaces for open CNCs. The result will lead users to an open system environment, where they will be able to reconfigure FA equipment freely according to their aims and how they use their manufacturing environment. Users will also be able to maintain and operate their system easily, and minimize life cycle costs.

3. OSEC-I Project

The OSE consortium was established in December 1994 for an open system environment by 6 Japanese companies, Toyoda Machine Works Ltd., Toshiba Machine Co., Ltd., Yamazaki Mazak Corporation, Mitsubishi Electric Corporation, IBM Japan, Ltd., Strategy Management Laboratory Corp. A summary of the OSEC-I project is as follows[2,4]:

Term of OSEC-I: from Dec.-1994 to Sep.-1995

Objective of OSEC-I:
1) Easy integration of desired CNC functions for users through openness of CNC.
2) Making the cost structure transparent by building a multi-vendor environment.

Characteristics of OSEC-I: 1) layered internal functions of blackbox of

conventional CNC and defined functions and roles for each layers.
2) defined interface protocols between layers
3) presented OSEL (OSE Language), which was a new NC language.

3.1 Reference Model for CNC Architecture

The conventional CNC is shown in Figure 3.1 and a reference model and the service functions of an open CNC are shown in Figure 3.2. The architecture of open CNC should be defined as a part of the manufacturing system to make process and data flow clear from design to manufacturing. In OSEC-I, it was classified into seven layers, consisting of an input layer, a layer for calculating trajectory of geometry, a controlling layer, etc. It contained CNC blackbox and defined the service functions of each layer.

3.2 Prototyping Systems

We defined the service functions at each layer and the interface protocols between layers based on the reference model, and developed three prototyping systems using the interface protocols. We verified the functions of OSEC architecture from the CAD systems to the machine tools and protocols, using the three prototyping systems shown in Figure 3.3.

(1) CAM Station: The NC data in OSEL format was generated by processing drawing data from an Autocad system. Monitoring was done in the CAM station.

(2) Station-1: A vertical machining center for machining parts was controlled by PC-based CNC.

(3) Station-2: Three servo motors with SERCOS specification were controlled with an OSEL interpreter by panel computer-based CNC.

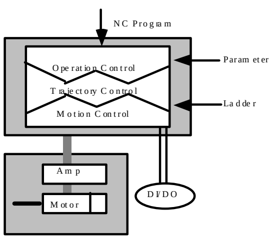


Figure 3-1 Conventional CNC

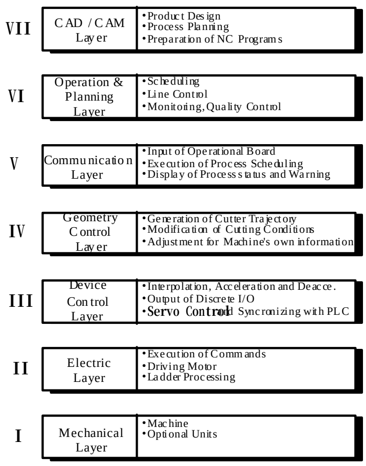


Figure 3-2 Reference Model and Services of Open CNC

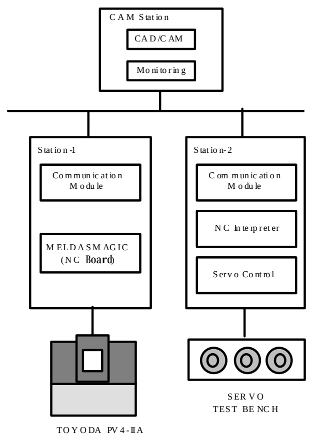


Figure 3-3 Prototyping Systems of OSEC-I

4. OSEC-II Project

4.1 Objective and Activity Overview

OSE consortium was started by 6 companies. Since its essential mission is to undertake research and development on controller architecture and a standard interface to realize an open manufacturing system environment in near future, OSE called for new participants, reorganized the consortium, and began an OSEC-II project from November

1995. The objectives of OSEC-II are

(1) to investigate architecture for open CNC and research its standardization from the long term view, exchanging information with similar projects internationally.

(2) to develop S/W and H/W for a prototype to verify architecture using advanced technologies.

Figure 4-1 shows the organization and the technical working groups of the OSE consortium. The only duty the OSE partners must perform is to join in the R&D. If this condition is satisfied, anyone can join. Now seventeen companies and one technical research institute are researching in the technical WGs.

(1) Steering committee: Plans and organizes the research direction, the IPR rules and publication, for example to JIMTOF.

(2) Executive committee: Works on actual jobs specified by the steering committee and discusses OSEC specifications.

(3) Technical WGs:

(3-1) MMI-WG: Researches a methodology for development of higher compatible and portable MMI software, based on a machine framework.

(3-2) OSEL-WG: Develops a feature base CAM system for machining planning, introducing the machining features consisting of geometry and attributes.

(3-3) Machine Control-WG: Develops machining software, an input of which is the OSEL programs.

(3-4) Device Control-WG: Develops the device drivers and the hardware of servo control and PLC (Programmable Logic Controller).

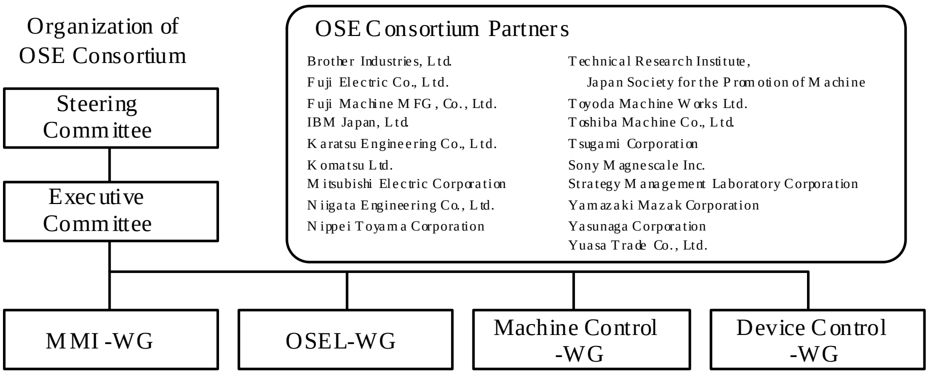


Figure 4-1 Organization and the technical working groups of OSE consortium

We had five general meetings to decide the OSEC specification for the architecture and protocol and each WG had between five and ten technical WG meetings up to August 1996. Since it is very difficult for partners from different enterprises to have frequent meetings, we usually discuss some problems by e-mail. A mail server is installed at the Strategy Management Laboratory Corp. site and some mailing lists for the committees and technical WGs also exist.

4.2 Architecture and Implementation

We defined a seven layer model for open controllers. Generally the controllers are realized by converting the abstract reference model to a functional architecture model and then finally to an implementation model. This implementation model can then be mapped to the H/W and S/W of microprocessors.

The reference model is suitable for getting a bird's eye view of the CNC system but it is not detailed enough to describe the actual system. To bridge this gap, we introduce the concept of function groups of CNC systems and define an architecture using the function groups in OSEC-II (Figure 4-2). Figure 4-3 shows some interfaces among the function groups. OSEC open CNC architecture consists of five function groups such as MMI function group, resource manager function group, language processor function group, machine control function group, and device control function group (servo control and PLC input/output).We think that the reconfiguration of machine tools will be achieved by modularizing the functions of the function blocks using S/W and H/W pairs. Figure 4-6 shows the architecture model with the protocol flows between the function groups. We break down the model so that we make the CNC implementation model. In the implementation design, it is necessary to consider the microprocessor, realtime OS, process/task, interrupt, and system and user programs. Finally we must consider DLL (Dynamic Linking Loader) for dynamic linking, an entry method to a configuration database of system, and system initialization.

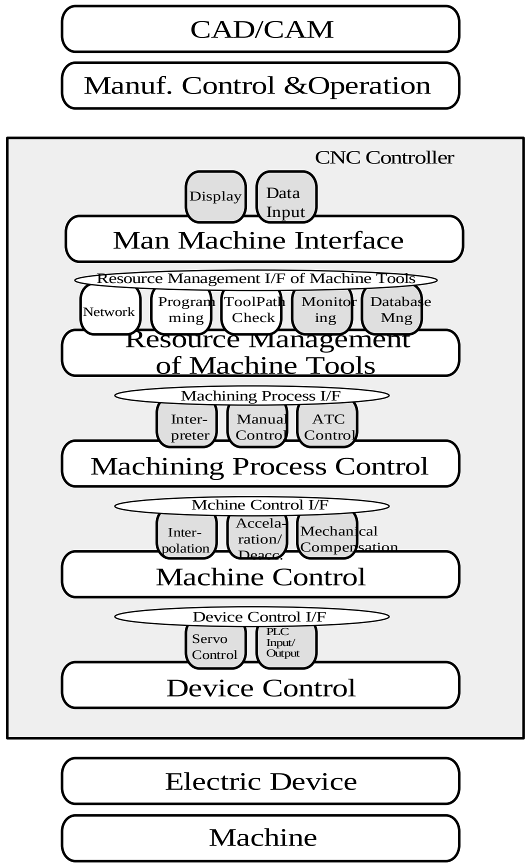


Figure 4-2 CNC Architecture

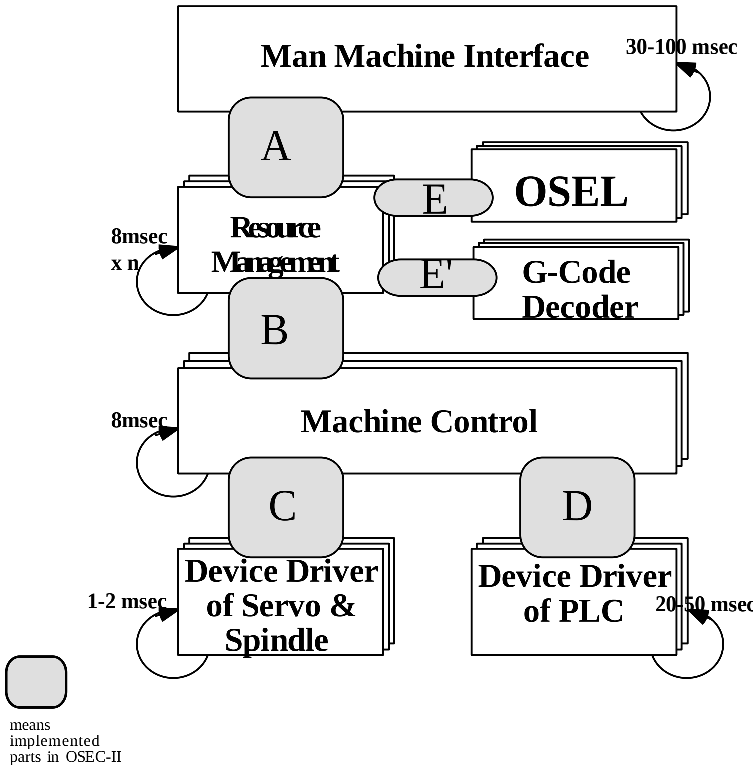


Figure 4-3 Interface between Function Groups

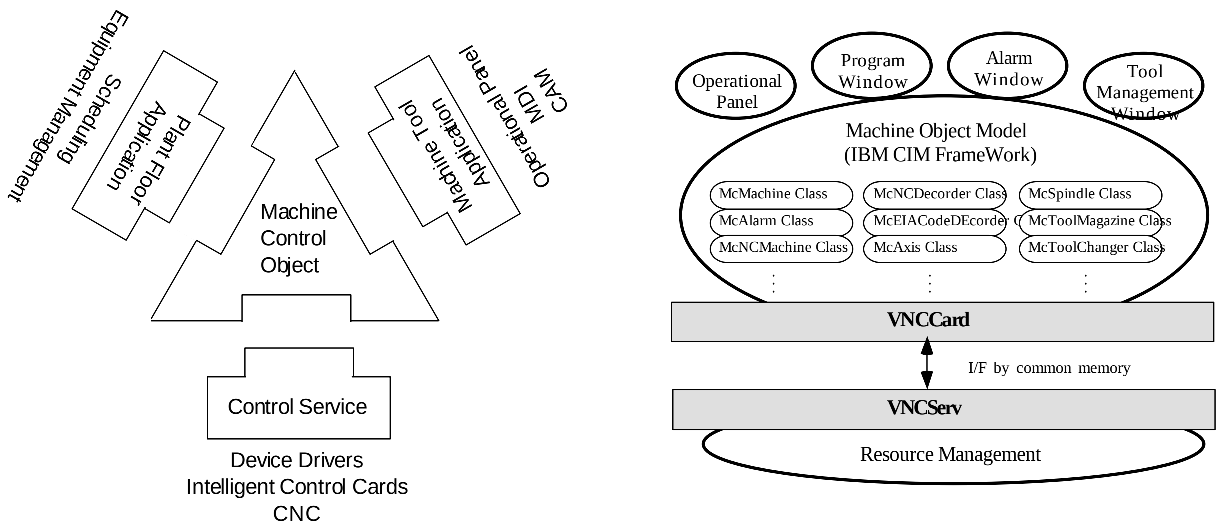
Each of the function groups can be distinguished by the realtime response requirement. The requirement is 30-100 msec for MMI during which time events are transferred from the outside to operators, 8msec x n for language processor and resource management which manages the state of machine tools based on the events from machine tools and S/W, 8 msec for machine control which controls motors and ladders, 1-2 msec for servo control, and 20-50msec for PLC input/output. The response type is divided into a random event type in the upper function groups and a cyclic event type in the lower function groups.

We are developing the prototype systems described in section 4.4, by designing the implementation model based on the functional architecture model. In the process, we define the interface protocols among the function groups shown in the appendix. We break down the function groups and derive the functions from them. This breakdown will allow us to partially develop open CNCs and third vendors with excellent technologies can easily enter the manufacturing business. Moreover, it is very easy to configure additional functions like those of CAM because function groups are modular.

Applying an object oriented methodology to the conversion process from the reference model to the implementation model is very useful in making the manufacturing functions modular. We try to apply the methodology partially to the MMI and OSEL function groups. To improve reconfigurability of manufacturing systems and machine tools, the methodology should be applied to all design processes in open CNCs. It is especially desired to aid the conversion design mentioned above consistently. This is one of the important problems to be solved for open CNCs.

4.3 Function Groups and Their Interfaces

4.3.1 MMI



(1) Machine Framework (2) Implementation of MMI Function Group

Figure 4-4 Machine Framework

An Operation panel, a program window, an alarm window, and a tool management window are developed based on a machine framework provided by IBM Japan using the VisualAge language as a GUI builder. Recently the manpower required for development of MMI software has tended to increase, in accordance with the progress in interactive computing and the higher performance of machine tools. The MMI-WG intends to abstract CNC controller and define an object model of machine tools in order to develop a very visible and portable GUI software. Figure 4-4 shows the machine framework for MMI.

4.3.2 Resource Management of Machine Tools

The resources of machine tools are operational panels, databases such as NC programs, tools, and parameters, NC language processors such as OSEL and G code decoder, software for machine control, servo motors, PLCs, and peripheral equipment. The resource management function group handles the events from these resources and manages the machine tool according to its mode. The abstracted CNC board (VNCCard) in the machine framework and the resource management server are linked by way of a common memory interface in the prototyping systems of OSEC-II (Figure 4-2). We are developing three systems and IBM Japan is developing a basic module of MMI. The partners of MMI-WG are developing MMI software using the basic module. MMI is used commonly in three systems and the machine tool makers develop MMI applications and the resource management module using the software.

4.3.3 New NC Language (OSEL)

Although conventional NC language in an EIA format can be easily used to operate machine tools, the language has illustrated its shortcomings in compatibility and readability. These shortcomings are due to the severe competition in macros and code definitions between NC makers. OSEC intends to develop a high level language for CNC, like the C language. This language is called OSEL, and can represent tool trajectories and machine tool operations such as tool exchange at a higher level. We will use OSEL not only to improve readability but also to develop high level CNC libraries through structured programming. We expect that third vendors with advanced technologies such as sensor-based adaptive control, will join the manufacturing business and will provide the additional CNC libraries which can give machine tools the added value (Figure 4-5 (1)). Machining classes in OSEL are divided into machining classes such as "face milling" and "pocket

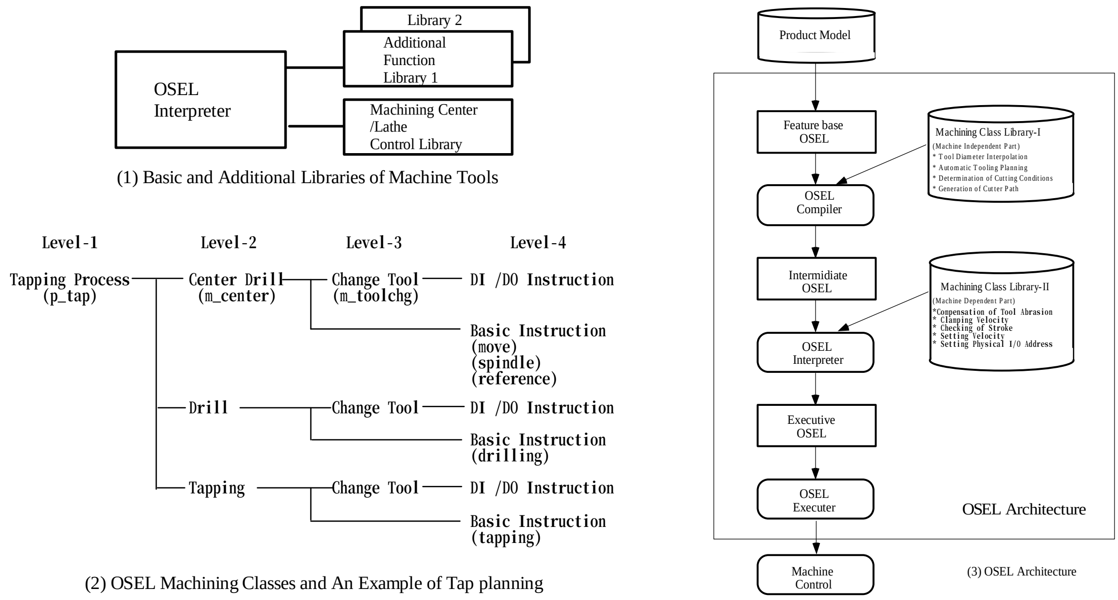


Figure 4-5 OSEL Concept and Architecture

milling", CNC classes such as "Move", "Line", and "Arc", and machine control classes such as "AxisStart" and "AxisMove". We developed an OSEL interpreter and the machine tool makers verified its effectiveness in the OSEC-I project.

Development of product model and STEP have been progressing so that engineering information about products will be able to be exchanged in product model format. OSEL-WG is trying to develop a feature-based CAM system. OSEL-WG intends to realize a processing algorithm for machine dependent information and also for adding the machining classes by introducing both a machine independent and a dependent processing part shown in Figure 4-5 (3).

4.3.4 Machine Control

The role of the machine control is to process the basic functions of CNCs.There is not only an OSEL interpreter but also a G-code decoder for compatibility with the existing user NC programs. After the machine control receives the machining commands from OSEL, it executes interpolation, acceleration / deaccelaration, generation of servo commands and input/output to ladders processing. Interface protocols such as device control, mode control, and axis control are defined in Table -2 of appendix.

4.3.5 Device Control (Servo Control and Input/Output of PLC)

There are international standards for the servo motors and PLCs: SERCOS (serial realtime communication system) specification between servo amplifiers and motors, and the standard software PLC products based on IEC 1131-3. When the users assemble the standard products in their systems, it is very difficult for them to develop software according to the specification. They prefer being supplied both the standard product and its device driver software for embedding the product in CNCs. Therefore we are defining the interface for the device drivers.

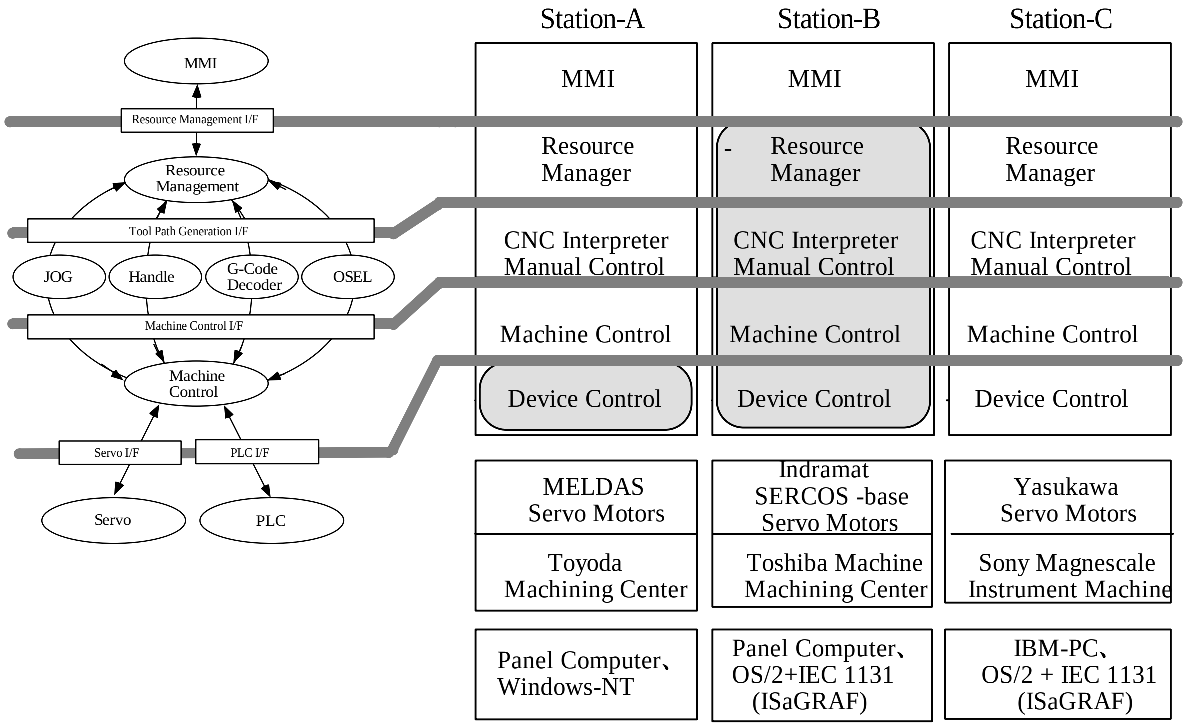
We intend to define a specification in order to build a business environment for multi-vendors in both servo control and PLC. We consider satisfying the basic functions, integrating the special functions by makers of servo devices and PLCs, and easily configurating without having to be modified during normal use

.

4.3.6 Prototyping Systems

Software modules corresponding to the function groups based on the CNC architecture have been developed and verified in the prototyping systems. Figure 4.6-(2) shows a comparison of implementation among the three systems. In this figure, the larger block describing station A, B, and C shows the CNC part and the shading and horizontal lines indicate the function groups using a common protocol among the stations. The shaded blocks mean that their corresponding functions are implemented by additional NC boards.

Station-A implements the device control function group with the NC board but implements other function blocks with PC-based software control. Station-B is a PC-based CNC and uses a software PLC for monitoring. It has servo drives with SERCOS specification so that it is a very close to products. Two stations control the machine tools and can machine the workpieces. Station-C is a pure software CNC and uses the software PLC for sequence control. It controls an instrument machine.



(1) CNC Architecture Model (2) Comparison of Implementation of Function Groups with Protocol Flow

Figure 4-6 Three Prototyping Systems and Implementation of Function Groups

In conclusion, by introducing the CNC architecture and the common protocols between the function groups, it is possible to

(1) exchange the OSEL and the G-code decoder,
(2) share the MMI software,
(3) build multi-vendor business environment for the servo drives, such as MELDAS, Indramat, and Yasukawa and PLCs

The open CNC can be developed under multi-OSs and multi-platform environments. Network computing technologies such as World Wide Web and JAVA can be evaluated for remote monitoring and remote machining management in the prototype systems.

5. Conclusion

With the aim of realizing open CNCs, we have presented the functional architecture and the interfaces among the function groups and confirmed reconfigurability of the different interpreters of the NC language, the common usage of MMI software, and the possibility of a multi-vendor environment for servo drives and PLCs. We expect that some specifications for the open CNCs will be standardized, and that an open manufacturing systems environment will be built by publicizing and discussing the results of the open CNCs projects. The results of OSEC's R&D will be made public in August 1996 and some of the machine tools with OSEC specifications will be exhibited by the machine tools makers in November 1996 at JIMTOF. The obstacles to open CNCs and an open manufacturing systems environment include architecture, advanced standardization of interfaces, and modularization of manufacturing functions. It can not be achieved without international collaboration among many companies. Our OSE consortium would like to hear many opinions and get a lot of advice about our activity.

Acknowledgement

Lastly the author gratefully acknowledges helpful discussions with Prof. Wada of Setsunan University, Mr. Noda and Mr. Asano of Toyoda Machine Works Ltd, Mr. Kanemoto of Toshiba Machine Co. Ltd, Mr. Inoue of Yamazaki Mazak Corp., Prof. Matsuka of Sannou College, Mr. Sawada of IBM Japan, Ltd, Mr. Maeda of Strategy Management Laboratory and the other OSE partners.

References:

1 Chrysler, Ford, GM, Requirements of Open, Modular Architecture Controllers for Applications in the Automotive Industry Version 1.1

2 OSE Consortium, OSEC: Open System Environment for Controller Architecture Draft 1.0, http://www.mli.co.jp/OSE/

3 R. Wada, Objectives of open architecture "OSE", the 26th '96 Machine Tools Engineering Meeting July-8th-1996.

4 Noda, OSEC -- Documents of panel discussion " How to join the movement toward open manufacturing systems", Forum on open manufacturing systems GO!, January-22nd-1996.

Appendix: Interface Protocols Between Function Groups

A part of the interface protocols between function groups is shown in the tables. Please refer to the OSEC-II report for the final protocols and their parameters because the protocols are under development now.

Table-1 I/F Protocol of Resource Management 1/2 (Resource Management <-> MMI)

Automatic Control Mode		Manual Control Mode	
osecMrOpen()	Open machine resource	osecMrJogAxisPlus()	
osecMrRunCommandGet()	Get run command	osecMrJogAxisMinus()	
osecMrEiaProgramGet()	Get program status	osecMrMagnification()	
osecMrEiaProgramSet()	Set program status	osecMrHandleAxis()	
osecMrWorkPosGet()	Get work position	osecMrHandleMag()	
osecMrRunStatusSet()	Set run status		
osecMrAlarmSet()	Set alarm	osecMrWorkPosSet()	Get work position
osecMrSingleGet()	Get single block	osecMrWorkOffSet()	Set work offset
osecMrBlkSkipGet()	Get block skip	osecMrMachPosSet()	Set machine coordinate position
osecMrOptStopGet()	Get optional stop	osecMrStopStatusSet()	Set stop status
osecMrStopStatusSet()			
osecMrCmdReset()			

Table-1 I/F Protocol of Resource Management 2/2 (Resource Management <-> NC Language)

OSEL		G -Code Decoder	
osecOselOpen()	Open file	osecEiaLoad()	Load Eia file
osecOselAlarmSet()	Set call-back routine for alarm	osecEiaSearch()	Search Sequence number
osecOselStart()	Start	osecEiaStart()	Start
osecOselStop()	Stop	osecEiaStop()	Stop
osecOselResume()	Resume	osecEiaClear()	Clear Eia decoder
osecOselFlush()	Flush command buffer	osecEiaAlarmCan()	Clear alarm
osecOselStep()	Single step	osecEiaCommand()	Command for test
osecOselSetStatus()	Set status		

Table-2 I/F Protocol of Machine Control

1.1 Machine Control Command		1.2 Mode Control Command (Option)	
osecMctrlOpen()	Open	osecMctrlModeSingleBlock()	Single block mode
osecMctrlClose()	Close	osecMctrlModeDryRun()	Dry run mode
osecMctrlReady()	Set Ready	osecMctrlModeStop()	Exact skip mode
osecMctrlFree()	Get number of free buffer	osecMctrlModeAxisCancel()	Axis control cancel mode
osecMctrllCancel()	Cancel buffer	osecMctrllOverCut()	Override cutting velocity mode
osecMctrlStart()	Start	osecMctrlOverPart()	Partial Override cutting velocity
osecMctrlStop()	Stop	osecMctrlOverMove()	Override rapid feed velocity mode
osecMctrllGetMode()	Set mode of getting data		

1.3 Axis Control Command		1.4 Axis Control Command (Option)	
osecMctrlAxisProgramOrigin()	Get origin of program coordinate	osecMctrlAxisArc()	Arc cutting
osecMctrlAxisWorkOrigin()	Get origin of work coordinate	osecMctrlAxisArcVector()	Arc cutting with norm. vector
osecMctrlAxisMove()	Move target position	osecMctrlAxisHelical()	Helical cutting
osecMctrlAxisLine()	Line cutting	osecMctrlAxisVelocity()	Move at specified velocity
osecMctrllAxisCps()	Cutting of continuous position	osecMctrllAxisTap()	Tapping
osecMctrlAxisCpsAdd()	Add continuous position data	osecMctrlSkip()	Skip
osecMctrlAxisWait()	Axis wait		
osecMctrlGetAxisVelocity()	Get axis velocity		