Standardization of Auxiliary Equipment for Next Generation CNC Machining

David Odendahl and Sid Venkatesh
Boeing Company
Seattle, WA

John Michaloski and Fred Proctor
National Institute of Standards and Technology
Gaithersburg, MD 20899, USA

Abstract

This paper presents the recent work of the Open Modular Architecture Control (OMAC) Machine Tool Working Group to support STEP-NC, which is a new standard for the exchange of comprehensive Computer Numerical Control (CNC) manufacturing data. Because of the myriad of CNC auxiliary equipment and countless RS274 dialects, OMAC members collaborated on a standard approach for legacy CNC program portability by defining an on-machine CNC Virtual Machine (VM) layer. We will discuss results and related issues from the use of the STEP-NC and a CNC Virtual Machine that was done in an OMAC pilot project at Boeing and NIST to handle auxiliary equipment functionality and allow portability of part programs by pushing non-standard operation onto the machine tool.

Keywords

Machine Tool, standard, Computer Numeric Control (CNC), RS274, STEP-NC, Open Modular Architecture Control (OMAC)

BACKGROUND

Today, subtle and often quirky idiosyncrasies exist between different Computer Numeric Control (CNC) machine tools, even from similar Original Equipment Manufacturers (OEM) machine tools running identical CNCs. A review of the auxiliary equipment handling from just one CNC vendor would find a divergent set of tooling conventions, offset handling, and kinematics configurations. Across vendors, the non-standard divergence is even more pronounced. Customizations of a machine tool CNC based on a procurement specification adds to the confusion. This disarray leads to productivity losses due to machine operation misunderstandings and programming confusion.

The predominant Computer Numeric Control (CNC) machine tool programming language for over the last half century has been RS274. During the course of this time, CNC vendors have augmented and extended the RS274 programming languages as they have seen fit to meet functionality needs and user requirements. This proliferation of modifications to the standard has led to hundreds of vastly different

RS274 dialects. This circumstance has been mitigated the Computer Aided Manufacturing (CAM) software vendors, who are burdened with maintaining hundreds of CNC-specific post-processors.

An ongoing initiative of the Open Modular Architecture Controllers (OMAC) Machine Tool Working Group [1], of which the authors are members, has been to champion “STEP-NC” as the next generation replacement for RS274. OMAC is an affiliate organization of the Instrumentation, Systems and Automation Society (ISA) working to derive common solutions for technical and non-technical issues in the development, implementation and commercialization of open, modular architecture control. The OMAC Machine Tool Working Group is a group of end-users, control vendors and machine tool builders who as part of their effort collaborate in the development of best practices and standards for the discrete parts industries.

STEP-NC is part of the ISO 10303 manufacturing data exchange standards, known as the Standard for the Exchange of Product Model Data, or STEP. The parts of STEP implemented in software systems are called Application Protocols (APs). STEP-NC is the colloquial term for ISO 10303 AP238, the “Application interpreted model for computerized numerical controllers”, a standard for the exchange of comprehensive CNC manufacturing data [2]. STEP-NC offers accurate and complete product definition data from product design all the way to the machine tool in the manufacturing pursuit to “design anywhere, build anywhere, and support anywhere.” Unfortunately, most CNCs do not provide native STEP-NC support so the OMAC Machine Tool Group has worked together to define a STEP-NC Virtual Machine functional interface to streamline adoption to the myriad of CNC RS274 native dialects.

In this work, a CNC Virtual Machine (VM) is a software layer that would allows a STEP-NC program to be translated into a standard RS274 [3] part program and be portable (for auxiliary equipment functionality) and to run correctly across machine tool CNCs. With a common virtual platform, part programs can be used on a number of CNC platforms as long as the virtual machine abstraction is used and has been implemented for the given CNC. Theoretically, this CNC virtual machine will eliminate the need to create separate versions of the same part program for different CNCs and machine tools. In reality, as we will argue later in our discussion section, this is much more difficult in application than in concept.

This paper gives an overview of the issues related to a CNC VM specification for Auxiliary Equipment to remedy the proliferation of RS274 dialects and ambiguity of these operations. The paper is organized as follows. In Section 2 we summarize the state of current part program generation for a machine tool and present the Virtual Machine concept, and then explain the details of our CNC VM as pertains to the tool change implementation. In section 3, we will discuss how we have implemented a CNC VM and issues of portability. We conclude with a short summary.

CNC VIRTUAL MACHINE

Machine Tool applications typically involve cutting a metal casting or block to create a part. Machine tools perform tasks such as milling, turning, boring, drilling, and tapping in order to machine simple to complex parts in many materials ranging from alloys to plastics. A machine tool will generally have three to five axes of motion, and a CNC computer to carry out the machining operations. Auxiliary Equipment includes devices, which supplements the machining operation such as spindles to hold the tool, tool changers, and tooling and fixturing. A sequence of CNC operations to machine a specific part is called a Part Program.

The initial motivation for the development of the CNC VM was due to the widely differing implementations of the RS274 standard. RS274 is the dominant standard for numerically controlled machines originally developed in the early 1960's and eventually evolved into ISO 6983 [4], an international standard. RS274 is a standard that provides many opportunities for customization especially given that the original standard left many M and G codes unassigned, which allows OEMS to create new and innovative programmable features without violating the standard [5].

Over the years, OEMS have extended the RS274 standard to add programming and control features clearly outside the RS274 scope, which is understandable, as new machining functions require it, and the urge to simplify operations, are ongoing. The original intent of RS274 was to provide a Numerical Control programming standard for machine tool operation. RS274 has succeeded in its original standardization goal but over the years the breadth and disparity of RS274 implementations has become marked. Our CNC VM specification constrains the proliferation of RS-247D programming language vendor customizations and extensions with a VM programming abstraction.

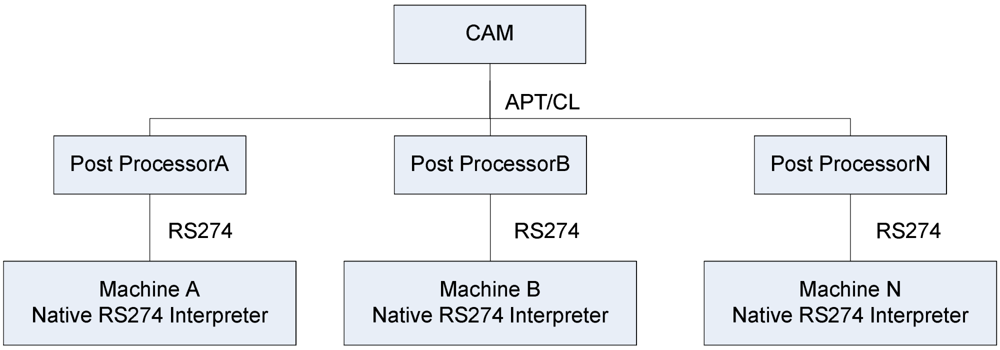

Figure 1 NC Program Generation

Post processing is now the principal means of mitigating and hiding the machine idiosyncrasies. Figure 1 outlines the generation of a legacy machine part program, where CAM (Computer-Aided Manufacturing) software, using a Post-Processor per machine, converts the CAD (Computer-Aided Design) part models into the required RS274 NC part program. Postprocessors are used to translate machining instructions (usually in Cutter Location or CL data [6] or Automatically Programmed Tool or APT format [7]) generated by a CAD/CAM or CAM system into RS274 “G-code” format that a CNC controller can understand and process [8]. For a post processor to work it must have complete awareness of what a machine tool is capable of and how it is supposed to be controlled, which is not always true. Problems are then uncovered during cutting tests, and fixed with ad hoc workarounds.

The first OMAC Machine Tool Working Group CNC VM objective was to develop a standard behavior in response to common CNC Auxiliary Equipment functions. CNC Auxiliary Equipment functions are the largest standards violator, and exhibit a behavior where most anything is allowed. Machine tool manufacturers/integrators implement auxiliary functions in general accordance with known conventions, but this can vary from machine to machine and from manufacturer to manufacturer. Again, this vagueness has been traditionally compensated for by the CNC post-processor developer, but this method is not ideal.

To develop the CNC Virtual Machine we used the largesse of RS274 to our advantage. Most CNCs have their own programming language extension to RS274 known as User Macros or Subprograms that allow standalone programs to be written that can be called from the main part program. These subprograms use the general-purpose programming features of variables, mathematical equations, and control structures, such as conditional branching and repetitive looping. With a Custom Macro program, blocks are written using variables and these variables can be changed each time the program is run. In addition, an RS274 Macro/Subprogram provides machine variables that give access to the state of the CNC. For example, macro programming allows the reading of state variables for position, feed, speed, machine limits, and tool offsets, among other machining parameters. Further, just like any programming, RS274 macro languages can be quite sophisticated, and offer the full complement of mathematical functions.

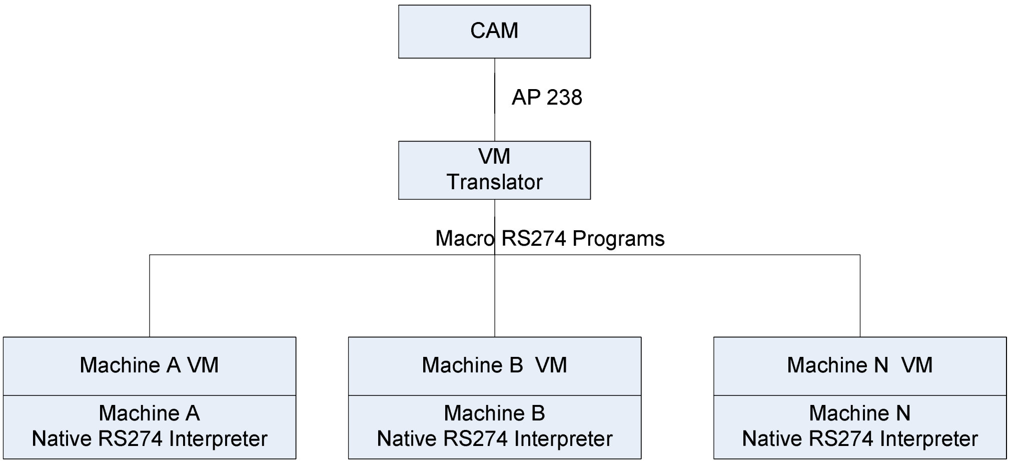

Figure 2 Legacy CNC Virtual Machine STEP-NC Program Generation

The salient feature that Subprograms and Macros offer is the layer of programming abstraction on top of the machine tool, hence the idea of the virtual machine. As examples, subprograms can be developed that update the coordinate system automatically, handle automatic clamping or standardize tool change handling. Thus, the virtual machine will be specified as a set of Subprogram calls that are implemented in the Native CNC RS274 dialect. Figure 2 outlines the part program generation based on a Subprogram-based CNC Virtual Machine. In this case, a CAM system takes the AP238 and uses a Virtual Machine translator that embeds generic macro calls instead of machine specific RS274 code within every part program that is theoretically more portable across machine tool CNCs.

Tool Change

To understand the background and rationale for CNC VM, a common "Change Tool" auxiliary function will be reviewed. In RS274, "M06" is a "Miscellaneous Function" that is responsible for changing the tool. In conjunction with the Tool change is the "Tool Selection" or "T" code. The RS274 specification explicitly states, "The T function code can be followed by 01 through 32 for tool selection, followed by 1 through 8 for tool length compensation, followed by 1 through 8 for cutter compensation." From [9], the T code is commonly used to rotate the machine's magazine (or tool storage device) to bring a tool into the ready or waiting position. An M06 exchanges the tool in the spindle with the tool in the ready position, (but, on some machines, the T word also does this). And an M19 rotates the spindle to align the key in the toolchanger arm with the keyway in the toolholder.

Summarizing, the issues related to Tool Change are as follows:

> **Machine Position before Tool Change**. Some machine tools allow a tool change as long as the tool change will clear the workholding setup. Others require the machine tool to move to its reference position in order to line up the toolchanger mechanism with the tool in the spindle.
>
> **Programming Method**. In a "Random access tool changer", the machine tool vendors allow a tool change at any time in the program. In this case the T# and M06 must be on separate lines in the RS274 program. In an "Integrated tool changer", the machine tool allows the tool change and the tool magazine carousel indexing to occur simultaneously, so that the required programming is for the M06 and T# to be in the same block.
>
> **Transformations.** Depending on the tooling, a variety of machining coordinate space functionality must be addressed.

This led us to the following Virtual Machine algorithm for the **Tool Change Operation**:

1. Retracting along the tool axis to a safe (safest) position
2. Transferring to machine coordinates for the tool change
3. Rapid motion to the tool change position
4. Changing the tool
5. Activation of tool length compensation
6. Transferring back to the workpiece coordinate system
7. Rapid motion to the safe (safest) position
8. Rapid motion to the original position

DISCUSSION

The genesis of the CNC Virtual Machine was due to a Boeing and NIST project to test the portability of STEP-NC Tool Center Program part programs [10]. For portability testing, the part selected was a half-scale National Aeronautics Standard (NAS) 979 5-Dimensional part (commonly known as the Circle Diamond Square or CDS) designed for runoff tests of new machine tools [11]. It was determined after a series of auxiliary function mishaps, that each machine tool would host a Virtual Machine defined as STEP-NC Virtual Machine workingsteps. Both parties programmed their own versions of the working steps for their machines, thus promoting portability by pushing machine dependent operation onto the machine. The collaboration between Boeing and NIST not only proved out the portability of STEP-NC, but that workingsteps simplified portability for legacy machines. These workingsteps were CNC subprograms called from the STEP-NC downloaded part program and have formed the basis for the Auxiliary Equipment CNC VM.

Figure 3 shows the code for a NIST tool change workingstep and Figure 4 shows the equivalent functionality adopted by Boeing for its tool change workingstep. The notable differences dramatically highlight the incompatibilities between each organization's machine tools as both working steps are coded for a Siemens 840D controller. Additionally, the CNC virtual machine can deal with diverse functional capabilities, as Boeing machines have more machine capabilities than the NIST machine tools, such as for example, the facility to specify motion tolerancing.

```
; NIST VERSION

PROC WSTOOL(INT AP238_TOOL)
N70  D0                 ;Tool length comp off
N80  G500               ;Workpiece offsets off
N90  TRAFOOF            ;5-axis xforms off
N120 T=AP238_TOOL       ;Get the new tool num
N130 M6                 ;Change the tool
N150 TRAORI(1)          ;Activate 5-axis xforms
N160 ORIWKS             ;We're in "part space"
N170 G55                ;Activate workpiece offset
N180 G90                ;Absolute programming
N190 D1                 ;Activate tool length compensation
N230 G1                 ;No more rapids
RET
```

Figure 3 NIST Tool Change Working Step

The CNC VM addresses standardization in the hope for improved program reliability and portability. In a perfect world, an NC part-programming standard would allow any program to run on any machine. The reality in the current machining milieu is that even if CNC programs could be “data-neutral”, they are not necessarily “process-neutral”. This means that for machines of a similar kinematics, the feeds, speeds and tooling that work for one machine tool can be completely wrong for another machine tool. Further, machining cultures can vary dramatically from small batch job shops who handle all aspects of design and fabrication to large production plants with 24/7 manufacturing with extensive staff for CAD/CAM/CNC programming, post-processing and machine operation.

```
; BOEING  VERSION

PROC WSTOOL(INT AP238_TOOL)
DEF REAL OLD_X, OLD_Y, OLD_Z
DEF REAL OLD_A, OLD_C

N10 OLD_X=$AA_IW[X]      ; x tool base coordinates
N20 OLD_Y=$AA_IW[Y]      ; y tool base coordinates
N30 OLD_Z=$AA_IW[Z]      ; z tool base coordinates
N40 OLD_B=$AA_IW[B]      ; a tool base coordinates
N50 OLD_C=$AA_IW[C]      ; b tool base coordinates
N60 M11                  ;Smog Hog off
N70 D0                   ;Tool length compensation off
N80 G500                 ;Workpiece offsets off
N90 TRAFOOF              ;5-axis transformations off
N100 G0Z=$AA_SOFTENDP[Z]-0.5A0 ;Retract to maximum Z
N110 G74C0.0             ;Resynch table
N120 T=AP238_TOOL        ;Get the new tool number
N130 M6                  ;Change the tool
N150 TRAORI(1)           ;Activate 5-axis transformations
N160 ORIWKS              ;We're in "part space"
N170 G55                 ;Activate workpiece offset
N180 G90                 ;Absolute programming
N190 D1                  ;Activate tool length compensation
N200 G0 C=OLD_C          ;Restore C position
N210 B=OLD_B             ;Restore B position
N210 X=OLD_X Y=OLD_Y     ;Restore X and Y position
N220 Z=OLD_Z             ;Restore Z position
N230 G1                  ;No more rapids
N230 TOL_5( .0012 )      ;Toolpath accuracy/speed
RET
```

Figure 4 Boeing Tool Change Working Step

As one assesses CNC VM for Auxiliary Equipment, part program portability must be analyzed for Local versus Global program compatibility. Local portability needs a close fit to the following “process-neutral” prerequisites:

- Machining culture - e.g., either job shop or production process
- Cutter technology - tooling materials of similar material and grade
- Tooling concepts - sizes and availability of tools is similar
- Machining techniques - e.g., use of high speed versus material hogging techniques.

Local portability refers to exchange between similar machines in the same organization. Global portability refers across to exchange organization/division using dissimilar equipment, and includes Local Portability. As the manufacturing trend toward globalization continues, incorporating more quantifiable process-related meta-information into the STEP-NC standard will be become even more important.

We are confident the CNC VM will fit easily into the existing CNC infrastructure with minimal effort and positive results. CNC vendors routinely provide a programming extension to their RS274 dialect, so that CNC VM implementation can leverage this functionality. For each machine tool, OEMs will provide the standard VM compliant routines to replace the current haphazard RS274 auxiliary equipment handling. Since the CNC VM is a programming abstraction on top of today's existing systems, there can be both short-term and long-term benefits. Short-term, the CNC VM will be of benefit to multiple industries, and for shops of all sizes. The long-term benefits will allow an emerging feature-based, knowledge-rich, STEP-NC part programming standard, a more stable CNC implementation platform, and offer even better portability.

To date, the OMAC Machine Tool Working Group has developed a CNC Virtual Machine for Auxiliary Equipment, Data Logging, and Safeguarding functionality and will be expanding our scope. CNC VM standardization will reduce confusion and make machine deployment more straightforward, as vendors with a better knowledge of their machine tool rather than users will implement the VM functions. Given standard CNC conventions, users can then expect to have a safer, more predictable machine operation across machine tools.

SUMMARY

Machine tool CNCs include Auxiliary Equipment in support of tooling, coolant, and other miscellaneous functions and their approach can vary greatly from machine to machine. We have presented a motivation for the use of a CNC Virtual Machine to handle Auxiliary Equipment, which offers an abstraction to standardize the great diversity of RS274 programming dialects and machine auxiliary equipment handling found on the factory floor. Collaboration between OMAC Machine Tool Working Group members on test parts successfully validated the CNC Virtual Machine portability.

DISCLAIMER

Commercial equipment and software, many of which are either registered or trademarked, are identified in order to adequately specify certain procedures. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology or Boeing, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

REFERENCES

[1] Open Modular Architecture Group (OMAC). www.omac.org.

[2] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2006 ISO 10303-238: Industrial automation systems and integration Product data representation and exchange Part 238:

Application Protocols: Application interpreted model for computerized numerical controllers. Geneva, Switzerland.

[3] ENGINEERING INDUSTRIES ASSOCIATION, RS274D, 1979. EIA Standard - EIA-274-D, Interchangeable Variable, Block Data Format for Positioning, Contouring, and Contouring/Positioning Numerically Controlled Machines. Washington, D.C.

[4] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 1982. ISO 6983/1 – Numerical Control of machines – Program format and definition of address words – Part 1: Data format for positioning, line and contouring control systems. Geneva, Switzerland.

[5] Herrin, G. E., 1989. “Why don’t we have part program compatibility?”. Modern Machine Shop.

[6] ELECTRONIC INDUSTRIES ASSOCIATION, 1992. ANSI/EIA-494-B-1992, 32 Bit Binary CL (BCL) and 7 Bit ASCII CL (ACL) Exchange Input Format for Numerically Controlled Machines. Washington, D.C.

[7] AMERICAN NATIONAL STANDARDS INSTITUTE, ANSI INCITS 37-1999, Programming Language APT, New York.

[8] Xu, X. W., 2004. Machine Tool Data in the STEP-NC Environment – Machine Tool Data for Postprocessing. White Paper.

[9] Lynch, M. “Efficient Tool Changes”. Modern Machine Shop Online.

[10] Venkatesh, S., Odendahl, D., Xu, X., Michaloski, J., Proctor, F., and Kramer, T., 2005."Validating Portability of STEP-NC Tool Center Programming”. In IDETC/CIE 2005 25th Computers and Information in Engineering Conference (CIE), ASME.

[11] NATIONAL AEROSPACE STANDARD, 1969. Uniform Cutting Tests - NAS Series: Metal Cutting Equipment Specifications. Washington, D.C.