... networking heterogeneous environments ... a commonplace in 2001 ... with information captured and processed at one node communicated to other very dissimilar nodes for perhaps surprising transformations and end uses ... in the Migration to 2001 ....

What's a Nice Computer Like the HP3000 Doing in a Place Like CAD/CAM?

Sam Boles, Member Technical Staff

HEWLETT PACKARD

Here's a simplistic mechanism to communicate quickly and briefly the basic concepts of CAD/CAM (Computer Aided Design, Computer Aided Manufacturing), and the networking of the diverse technologies involved.

We start by building an IMAGE database on the HP3000 with name and other fields. We extract names from the database and network them over to the HP9000 where the ASCII string is sized, scaled, centered vertically and horizontally and transformed into tool path geometries for downloading to a 3-D CNC milling machine to produce name plates.

Then it's back from the HP9000 to the HP3000 for engineering specs documentation to wrap up this "CAD/CAM in microcosm," with a glimpse of a new methodology that can increase productivity by an order of magnitude in product development.

This unusual heterogeneous network result for an otherwise ordinary IMAGE database on the HP3000 can give you an informal introduction to the world of CAD/CAM.

... evolutionary perspective: it wasn't our size, strength or speed -- but our tools that put us on this end of the leash ....

Of all the animals whose ancestors crawled up out of the swamp that day, we're ***not the most imposing***.

Look at our ***size***. Lots of animals are bigger. Elephants. Whales. Giraffes. To name just a few.

Look at our ***strength***. Ever arm-wrestle a gorilla? Ever wonder why Budweiser uses Clydesdales to pull the wagon instead of a team of Arnold Schwarzenegger's?

And ***speed***. Do you think dolphins aren't allowed in the Olympics because they're not ***fast*** enough? How many gold medals do you think Spitz would've won that year if there'd been one even ***average*** dolphin in the swimming events? No, it wasn't our ***size, strength or speed***. It was our ***tools*** that made the difference. That's why the leash is around our ***hand*** and ***not around our neck***.

Sure, the ***opposing thumb*** was handy. And being able to balance, however precariously at times, on ***two feet*** was a help. But it was our tools that really made the difference.

Tools meant we could ***see things that weren't there***. We could look at a tree and see a lever, a club, an axle, a torch, a spear, a raft, a shovel handle, even a baseball bat, for those who would take the world serious . . . .

. . . CAD/CAM as a transformation of data from the abstract to the concrete . . . a simple case . . . .

As the centuries passed our tools got more refined. One of these tools in the long heritage is CAD/CAM (Computer Aided Design, Computer Aided Manufacturing). Let's look at a simplistic example of CAD/CAM and see some of the basic principles at work. A "***CAD/CAM in microcosm.***"

Let's start with an IMAGE database on the HP3000. It can be a personnel-type database with people's names in it. Our data capture can be ordinary, too: thru a terminal where we touch the key with the letter "A" on it and thru some magic a nobit-nobit-nobit-bit-nobit-bit ends up in the memory of the 3000. A binary group in an NMOS memory array that can be interpreted thru the ASCII coding scheme as the letter "A."

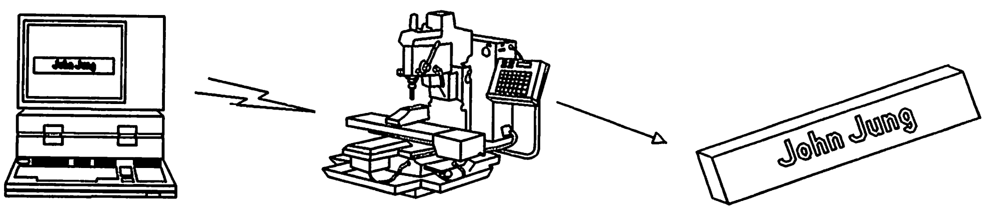
Now this is ordinary everyday commercial data processing on a computer. Nothing particularly marvelous about that. Well, marvelous, yes -- but we do it all the time everywhere so we've gotten used to the marvel.

Next let's extract some names from the database and communicate them via an RS232 ***network to another computer***, an HP9000. We can use QUERY to do the extraction and formatting and off-the-shelf drivers to send the name data serially over a twisted pair at 9600 baud to the 9000 running in terminal emulator mode. This is our CAD engine.

Now let's look at the next node in the network, that provides the CAM part of this "***CAD/CAM in microcosm.***"

. . . the CAM node in the network . . . a 3-D CNC machine tool with RS232 interface . . . .

The specific CNC (Computer Numeric Control) machine tool we'll focus on here is the ***Dyna 2400 milling machine***.



It's made by Dyna Electronics in Santa Clara, California, and represents some of the more advanced CAM technology available today.

The machine is designed to mill small, high-precision parts such as those typical of the electronics industry. The three step motors that provide the x-, y- and z-axis movement on the 2400 have a resolution of 1/10000 of an inch. The travel is about 6 inches in the x-axis, 5 inches in the y-axis and 4 inches in the z-axis. The maximum feed rate is about 30 inches a minute with synchronous and asynchronous control in each axis for full 3-dimensional capabilities, such as machining a hemisphere. The spindle can be throttled up to 10,000 rpm.

The on-board microprocessor is programmable in stand-alone mode or may be downloaded from a host via RS232 interface with line, subroutine and full-program granularity.

The instruction set includes the common control and vector move commands, but in addition is enriched with polar coordinate commands which simplify programming and provide powerful 3-D capabilities.

The machine's footprint is about 2 feet by 2 feet, with a height of about 2 feet and weight around 220 pounds. This gets it close to "***desktop***" class and enables comfortable positioning on a engineer's workbench for some interesting methodology implications as we'll see later.

. . . the "CAD" transformation of the ASCII symbol into 3-D CNC tool path geometries . . . .

Meanwhile, back at the HP9000, we have our ASCII symbol: someone's name or similar data. The data could just as easily be a ***vertex list for polygonal graphics or other 2- or 3-dimensional geometric descriptions with vector and/or polar coordinates***. For our CAD/CAM example, we've chosen an ASCII symbol to give a simplistic CAD illustration of transforming an alpha character into a tool-path geometry. Here's a narrative of what we do:

I. Abstract: This module is the CNC milling machine (Dyna 2400) stager and driver for name plates to illustrate CAD/CAM capability via a simplistic example. The process includes the design implications of transforming text to tool path geometries with sizing, translation and scaling; the driving of the milling machine with vector and polar coordinates as well as other control commands is an example of relatively sophisticated computer-aided manufacturing.

   The module loads external tables of width, key and geometries for tool path generation, gets a name/string, scales/centers/gens tool path commands for the machine, downloads the path program via RS232 link, and graphically simulates the milling action.

II. Input
   1. External table of subroutine reference key and size
   2. Tool path geometries in Dyna format
   3. Name(s) from HP3000 IMAGE database via terminal emulation

III. Processing

```
   1. Initialize flags, counters
   2. Load path, subroutine, sub index and width tables
   3. Get name/string, scan, scale, center
   4. Gen tool path program for download to CNC
   5. Download tool path program from 9000 file to CNC
   6. Scan/parse, simulate tool path with graphics CRT

IV. Output
   1. Graphic (CRT) replica of name plate, scaled and centered
   2. Temporary disc file ('TEMPPROG') of Dyna tool path code
   3. RS232 transmission of TEMPPROG to Dyna programmable controller
   4. Graphic (CRT) generation of tool path to simulate milling

 V. Techniques/Controls/Considerations
   1. Subroutine/width table (default 'MMTBL')
       Format:  123456789
                A 67 2.45
                A=subject letter; 67=subroutine #; 2.45=width
   2. Tool path table (default 'MMLTR')
       Format: 123456789ABCDEFG
               851 GR a-180.000
               851=Dyna program line number; GR=Go Relative (to
                local zero/tool location; a=angle of polar
                coordinate;-180.000=angle size in degrees
    . . .

   3. Data structures and linkages
     Ordinal value of character in string is offset into
       |
       |-> subrtntbl (0..255 INTEGER)----------------->|
       |    program gen, offset set here is used to    |
       |    match subroutine # from pathtbl to build   |
       |    path index                                 |
       |                                               |
       |-> widthtbl (0..255 REAL)                      |
       |    letter width in base mm unit to size for   |
       |    scaling                                    |
       |                                               |
       |-> pathidx (0..255 INTEGER) <------------------|
             index to path table, points to start
       |<-- of geometries for a given letter
       |
       |-->pathtbl (1..16232 CHAR)
            tool path commands with variable number
            of 16-byte records comprising the sub-
            routine for the geometries of a given
            letter
```

For each letter in the alphabet, we design a scalable, relocatable tool path. For simplicity we use a Helvetica font style (that is, no serifs as in a Roman font), which blends aesthetically with a 1/8 or 1/16 inch ball end mill in the travel range available. To further enhance the aesthetics we use relative instead of absolute positioning at character termination to enable a proportional Helvetica font.

Here's an example, using the letter "B":

```
093 SUB        12
094 GO Z-   0.200
095 GR Y    5.000
096 GR X    1.100
097 ZERO     XY
098 ZERO AT
099     Y-  1.200
```

```
100 GR a-180.000
101 GRcX-  1.100
102 ZERO AT
103     Y-  2.500
104 GR a-180.000
105 GR X-  1.100
106 Z>C
107 GOfX   1.300
108 SUB RETURN
```

This shows you in the Dyna CNC vernacular, some of the mechanics of getting your CAD design transformed into something tangible like metal, plastic or wood.

Here's a loose translation:

The SUBroutine is #12 (93). GO absolute to -0.2 units in the z-axis (94). Note that the spindle is assumed "ON"; this is done in the initialization. The tool is assumed "clear" of the workpiece; see the end of this subroutine for the convention.

Go Relative (relative to the current tool location) +5 units in the y-direction (95). Note that the tool is assumed to be at the lower left corner of the "cell." This gives the left side of the "B." Go Relative 1.1 units in the x-direction (right) (96). This gives the top of the "B."

Create a local ZERO reference point at the XY coordinates where the tool is currently located (97). Create a local ZERO reference point AT a coordinate -1.2 units from the current reference point in the y-direction, but with the x-coordinate unchanged (98-99). This establishes the center point for the upper arc in the "B". The radius is from this point to the current location of the tool. Go Relative (relative to the last two points with the radius demarked by those two points) in a clockwise (-) direction for an angle of 180 degrees (100).

Go Relative and come back (c) 1.1 units to the left in the x-direction (101). Cut the bottom arc of the "B" (102-104). Go Relative 1.1 units to the left (-) in the x-direction (105). Move the tool clear of the work piece in the z-direction (106). GO fast (f) in the x-direction to positive absolute 1.3 units (107). This puts the tool at "absolute" x of 1.3 wrt the local ref x of 1.1 for a cell width of 2.4 units. This is the lower right corner of the cell, since the tool y-location was at the bottom of the lower arc when last heard from. RETURN to the next sequential instruction of the caller (108).

Notice the arbitrary "user unit" of measure. This is the base for establishing the correct relativities and proportions. As we'll see later this enables ***automated scaling*** and centering in the horizontal and vertical for varying size names and other symbols. This translates into simplified and automated set-up.

The convention of starting the the lower left of the character "cell" and ending at the lower right enables an aesthetically pleasing "proportional font" in which the letter "m" is wider than the letter "i."

One of the powerful features of the Dyna CNC instruction set is the use of ***polar coordinates***, as we see in the code for the letter "B" above. We're able to establish a reference point without tool movement from which to trace an arc with the 1/10000 inch resolution step motors in 3 dimensions. Alternatively we would have to do something like the segment fabrication we do in the CRT tool path simulation routine:

```
IF coodbuf1 = 'a' THEN BEGIN
  {to simulate the Go Relative polar (angle)
    coordinate, build the polygon around a point
    rotated clockwise or counterclockwise with
    5 degree resolution}
  anglegr:=xyz;
  anglesteps:=ABS(TRUNC(anglegr/theta));
  FOR q:=0 TO anglesteps DO BEGIN
    IF anglegr < 0 THEN BEGIN
      xpolar:=(xcur-xloc)*costheta+(ycur-yloc)*sintheta;
      xpolar:=xpolar+xloc;
      ypolar:=-(xcur-xloc)*sintheta+(ycur-yloc)*costheta;
      ypolar:=ypolar+yloc;
    END
```

Since the names and other symbols can be of varying lengths, and since we have a physical limitation of about 6 inches travel in the x-axis, we use the HP9000 for CAD scaling of the name.

We size the name by scanning character by character, calculating the width in basic "user units." Then we scale the horizontal as the ratio of the x-axis travel limit to the calculated width. To maintain the original aspect ratio, we scale the y-axis with the same factor.

We are able in our original set-up calibration to position the spindle in the physical center of the work piece, so once we have the scaled x- and y-dimensions of the symbol, we can use half of the values to compute and execute the offset for the lower left corner. This way a simple physical calibration of the x-, y-, z-axis and spindle clearance can serve multiple runs on variable-length symbols.

It's beyond the scope of our "***CAD/CAM in microcosm***," but let's touch on some other issues you might want to address in more advanced problems.

You might want to input a material identifier that your CAD station could use to look up spindle rpm and tool feed-rate specs to download to the CNC device. You might want to programmatically do tool changes to handle more intricate cuts. You might want to control the injection of tool lubricant. You might want to coordinate the action of a loader/unloader robot. These things are within the capability of today's CAD/CAM technology.

. . . desktop computing . . .
. . . desktop drafting . . .
. . . now desktop machining for a major gain in productivity . . . .

In 1968 the HP9100 came to the engineer's workbench. A programmable calculator. With the HP9100, engineers no longer had to key in the algorithm each time they wanted to change a few variables. ***Desktop Computing.***

This enabled engineers to get their sketch to the drafting department faster. But that was an ***interface***. A time-consuming interface. So personal CAD stations emerged to produce finished drawings faster. ***Desktop Drafting.***

This enabled engineers to get their finished drawing to the model shop faster. But that was an ***interface***. A time-consuming interface. So computers and CNC machines teamed up to give engineers CAD/CAM at their fingertips. ***Desktop Machining.*** From "***art to part***" in minutes or hours instead of days or weeks.

. . . CAD/CAM engineering specs controlled with the data-words-graphics integration power of the HP3000 . . . .

For a professional grade documentation package that can leverage the investment in CAD/CAM, the scene can return to its starting point on the HP3000.

The 3000 has text and document preparation software such as TDP that can do ***typesetting*** at the quality level you see in this paper.

The ***graphics database*** on the HP9000 can be transformed and networked to the HP3000 where the vector form can be converted to raster for laser printer compatibility. With appropriate scaling and composition the documentation package can have a typeset "look" while retaining the ***flexibility*** of computer update.

Here's an example of graphics done with an HP9000 CAD system, ported to the HP3000 and integrated with text:

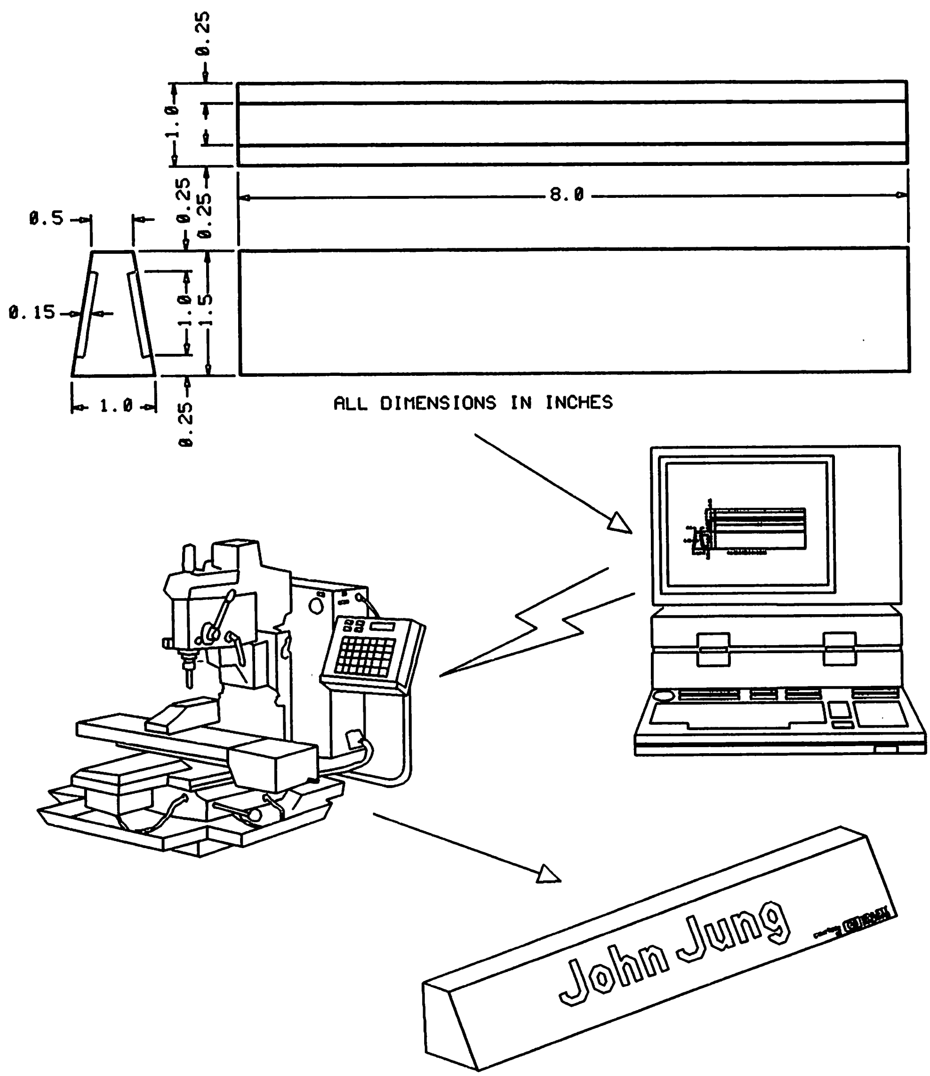


Figure 2. Our "***CAD/CAM in microcosm***" includes Engineering Data Control and documentation packaging thru a team effort of the HP3000 and HP9000 computers.

Epilogue ....

You're accustomed to the transformations and linkages for writing your data on disc, tape and paper. Here you've seen the transformations and linkages for writing your data on wood, metal and plastic. You've seen an ordinary ASCII string in an HP3000 IMAGE database networked to an HP9000 where it was transformed to tool path geometries executed by a CNC machine tool coupled with the HP9000. And you've seen the descriptive geometries of an HP9000 CAD system integrated with text on the HP3000.

This glimpse of CAD/CAM may help your understanding and appreciation of the other marvels we'll be seeing in our Migration to 2001.

About the Author ....

Sam Boles *is a Member Technical Staff in the Hewlett-Packard Information Software Operation in Cupertino, California. With HP since 1976, his computer experience started back in the AUTOCODER days of the 1401/1410, migrated thru the 360/370 era, and now focuses on next-generation operating system software. Sam received his MS at UCLA in Information Systems.*

sebiug27 2105 27jan86