

Computer-aided Engineering and Manufacturing

Teodora Mcfarland

First Edition, 2012

ISBN 978-81-323-3421-7

© All rights reserved.

Published by:

Research World

4735/22 Prakashdeep Bldg,

Ansari Road, Darya Ganj,

Delhi - 110002

Email: info@wtbooks.com

Table of Contents

Chapter 1 - Abaqus

Chapter 2 - Computer-integrated Manufacturing and Digital Read Out

Chapter 3 - STEP-NC

Chapter 4 - Numerical Control

Chapter 5 - Material Requirements Planning and Direct Numerical Control

Chapter 6 - G-code

Chapter 7 - Computer-aided Manufacturing

Chapter 8 - CAD/CAM Dentistry and Integrated Computer-Aided
Manufacturing

Chapter 9 - WorkNC

Chapter 10 - CATIA and HyperSizer

Chapter 11 - MPDS4

Chapter-1

Abaqus

Abaqus FEA (formerly **ABAQUS**) is a suite of software applications for finite element analysis and computer-aided engineering, originally released in 1978. The name and logo of this software is derived from abacus and the Greek word, “abax” (ἄβαξ), meaning “board covered with sand”.

The Abaqus product suite consists of four core software products:

1. *Abaqus/CAE*, a **Computer-Aided Engineering** software application used both for design and modeling machinery (pre-processing) and visualizing the Finite element analysis result
2. *Abaqus/CFD*, a **Computational Fluid Dynamics** software application which is new to Abaqus 6.10
3. *Abaqus/Standard*, a general-purpose Finite-Element analyzer that employs implicit integration scheme (traditional)
4. *Abaqus/Explicit*, a special-purpose Finite-Element analyzer that employs explicit integration scheme to solve highly nonlinear systems with many complex contacts under transient loads.

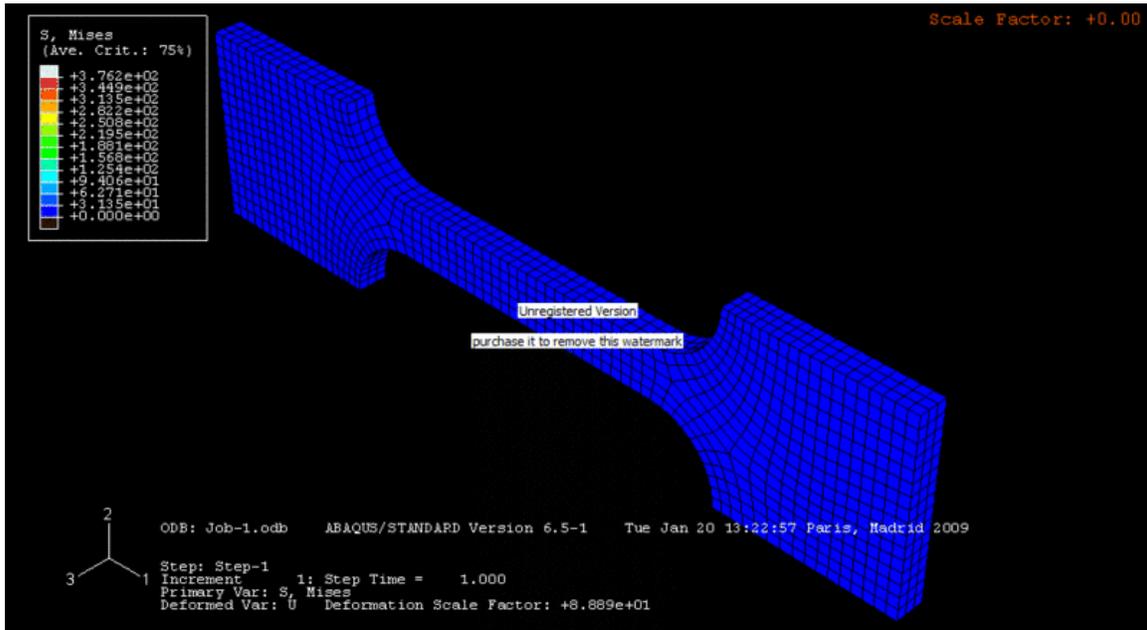
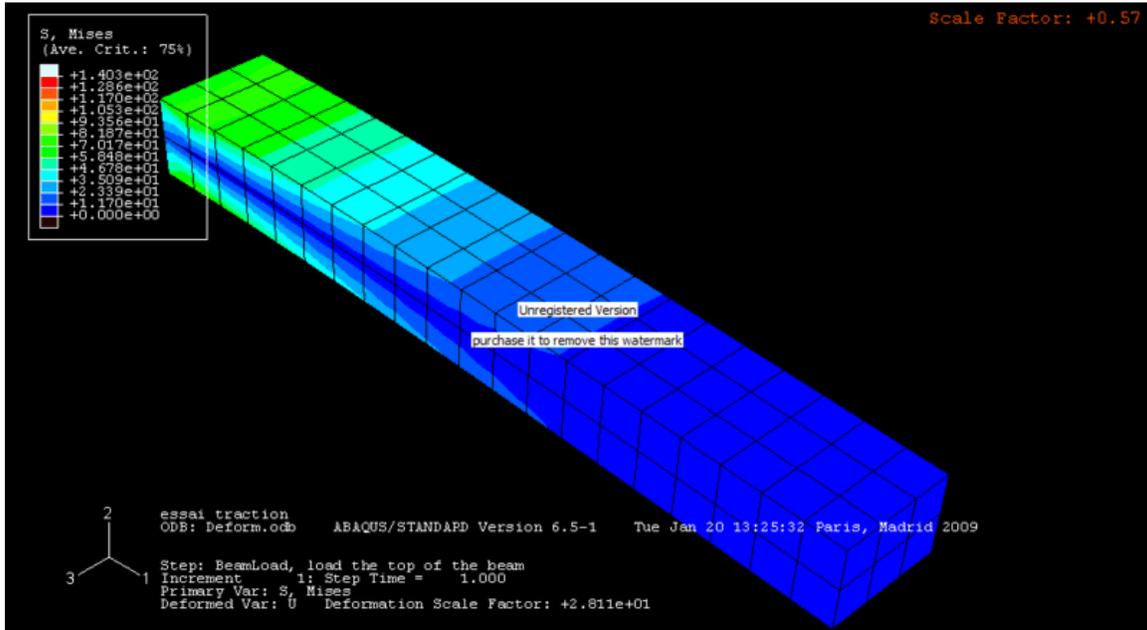
The Abaqus products use the open-source scripting language Python for scripting and customization. Abaqus/CAE uses the fox-toolkit for GUI development.

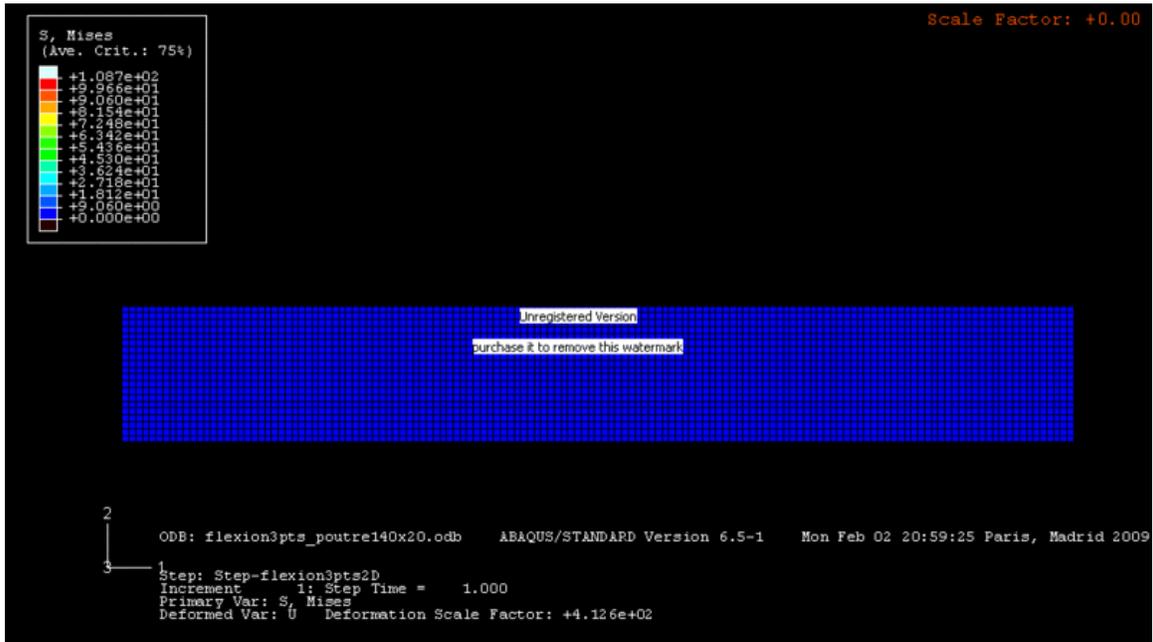
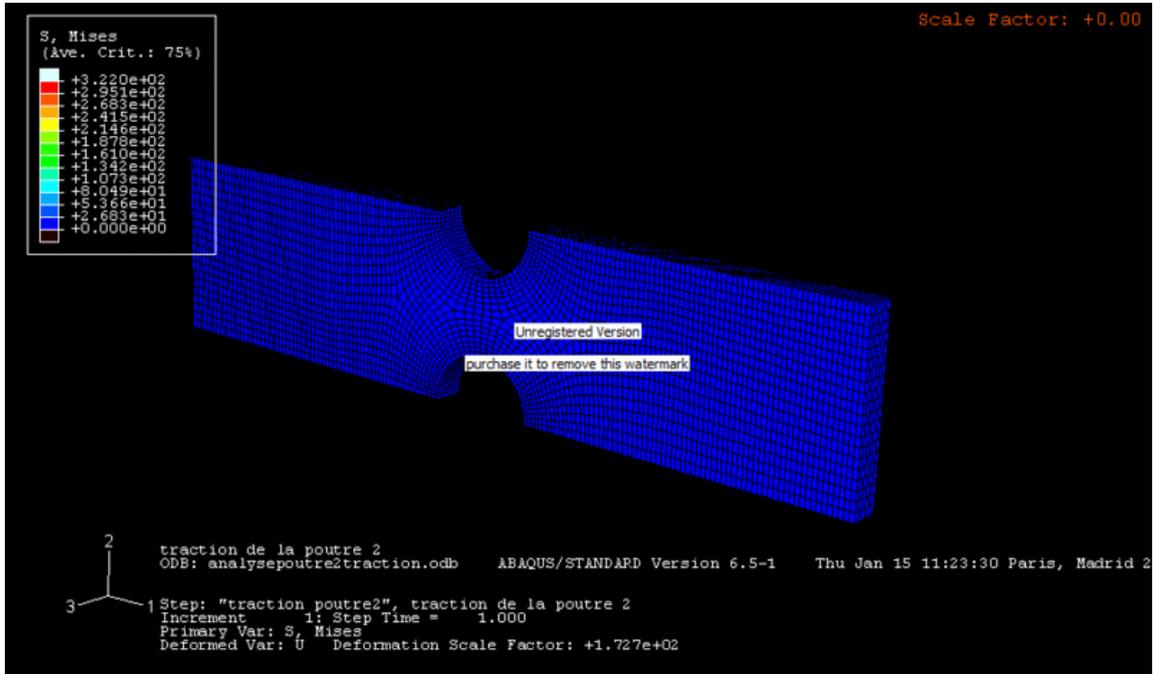
Applications

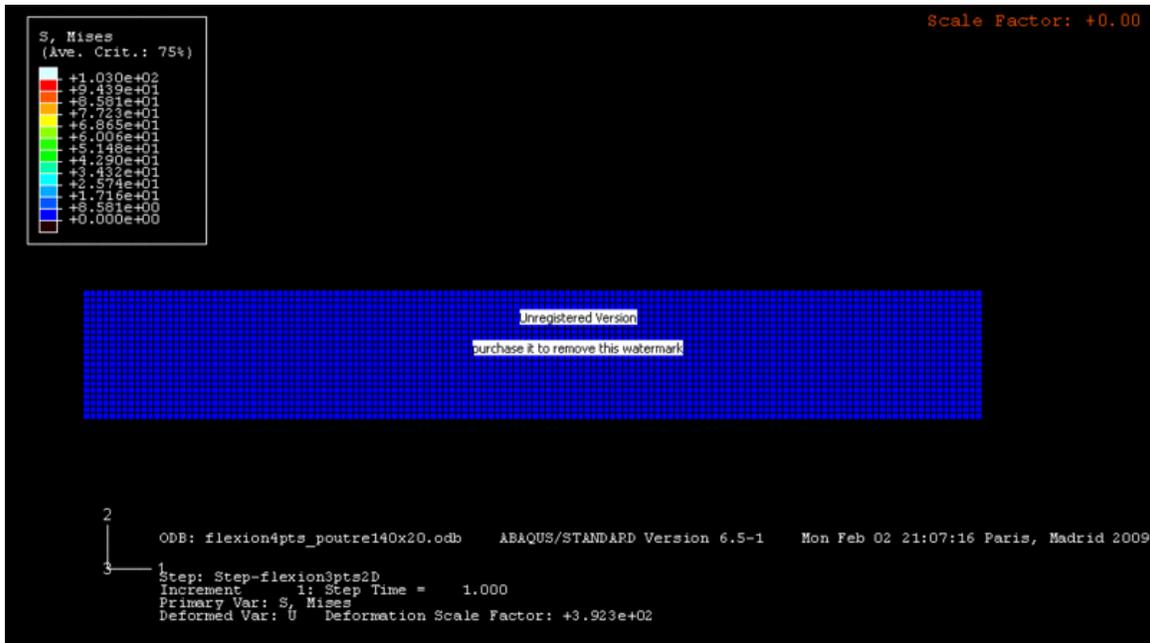
Abaqus is used in the automotive, aerospace, and industrial products industries. The product is popular with academic and research institutions due to the wide material modeling capability, and the program's ability to be customized. Abaqus also provides a good collection of multiphysics capabilities, such as coupled acoustic-structural, piezoelectric, and structural-pore capabilities, making it attractive for production-level simulations where multiple fields need to be coupled.

Abaqus was initially designed to address non-linear physical behavior; as a result, the package has an extensive range of material models such as elastomeric (rubberlike) material capabilities.

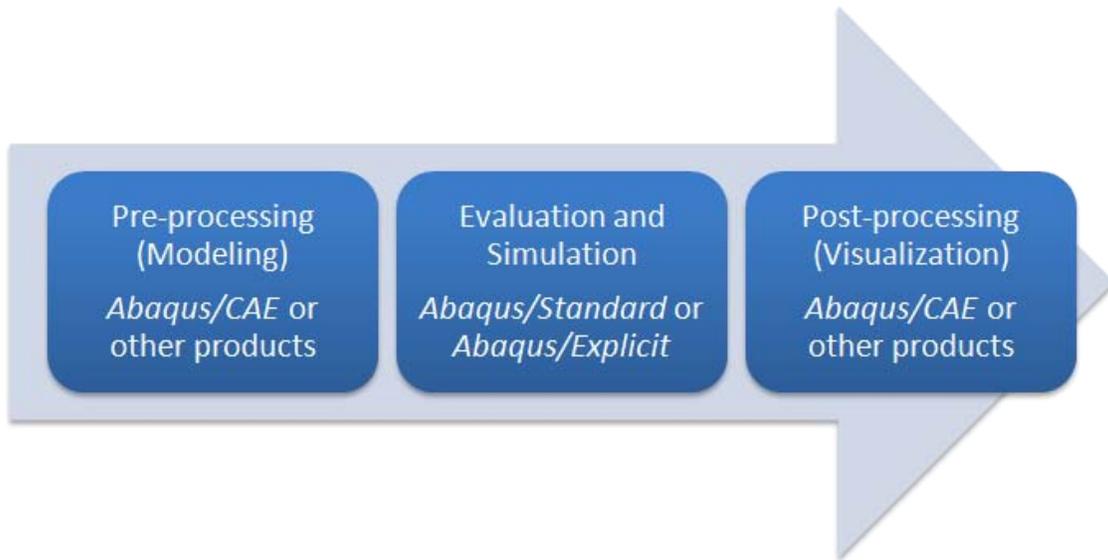
Here is some examples







Solution Sequence



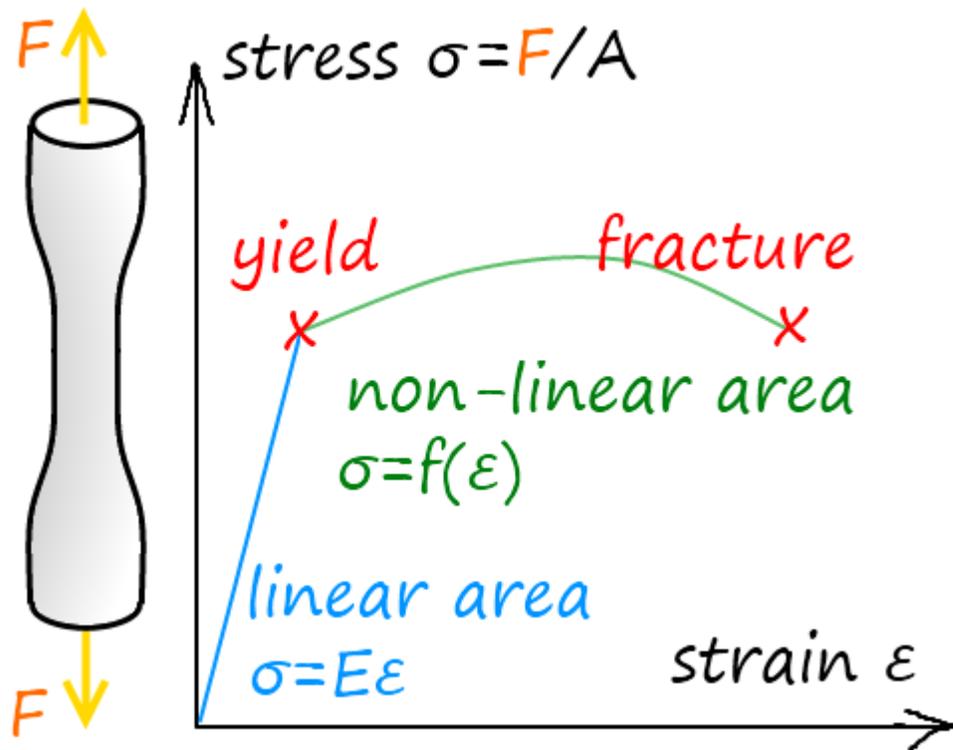
Abaqus FEA software products used in Finite Element Analysis and their order of use.

Every complete finite-element analysis consists of 3 separate stages:

- *Pre-processing* or *modeling*: This stage involves creating an input file which contains an engineer's design for a finite-element analyzer (also called "solver").
- *Processing* or *finite element analysis*: This stage produces an output visual file.
- *Post-processing* or generating report, image, animation, etc. from the output file: This stage is a visual rendering stage.

Abaqus/CAE is capable of pre-processing, post-processing, and monitoring the processing stage of the solver; however, the first stage can also be done by other compatible CAD software, or even a text editor. Abaqus/Standard, Abaqus/Explicit or Abaqus/CFD are capable of accomplishing the processing stage. Dassault Systemes also produces *Abaqus for CATIA* for adding advanced processing and post processing stages to a pre-processor like CATIA.

Solvers Comparison



Basic static response of a specimen under tension

The following is a comparison between the solver capabilities of Abaqus/Standard and Abaqus/Explicit.

| Feature | Common | ABAQUS/Standard only | ABAQUS/Explicit only |
|------------------|---------------------------------|---|--|
| Element library | Comprehensive no limits | | only elements appropriate for explicit solutions |
| Material models | Comprehensive only yield models | | yield and fracture models |
| Solution methods | | Implicit Integration needs solve multiple coupled equation Using the K Matrix | explicit integration step by step using small time steps sometimes not stable |

| | | | |
|------------|-----------------|-------------------------|------------------------------|
| | | (F=KX) | |
| | | Stable | |
| Required | | repetitive calculations | no repetitive calculation |
| Disk Space | | likely | normal |
| | | takes a lot of space | |
| | | | Can solve |
| | | | Optimal. even if highly |
| | | | non-linear |
| Types of | Linear: | Can solve | Optimal. even for complex |
| Problems | non-linear: | Can solve | and varying conditions |
| | Contact* : | Can solve if simple | Optimal under transient*** |
| | usual systems** | Optimal under steady*** | loads like Impact, Pulse and |
| | | loads | Explosion |

Notes

^* The more complex the contacts become, the more repetitive calculations ABAQUS/Standard has to solve, and the more time and disk space needed; ABAQUS Explicit is the optimal choice in this case

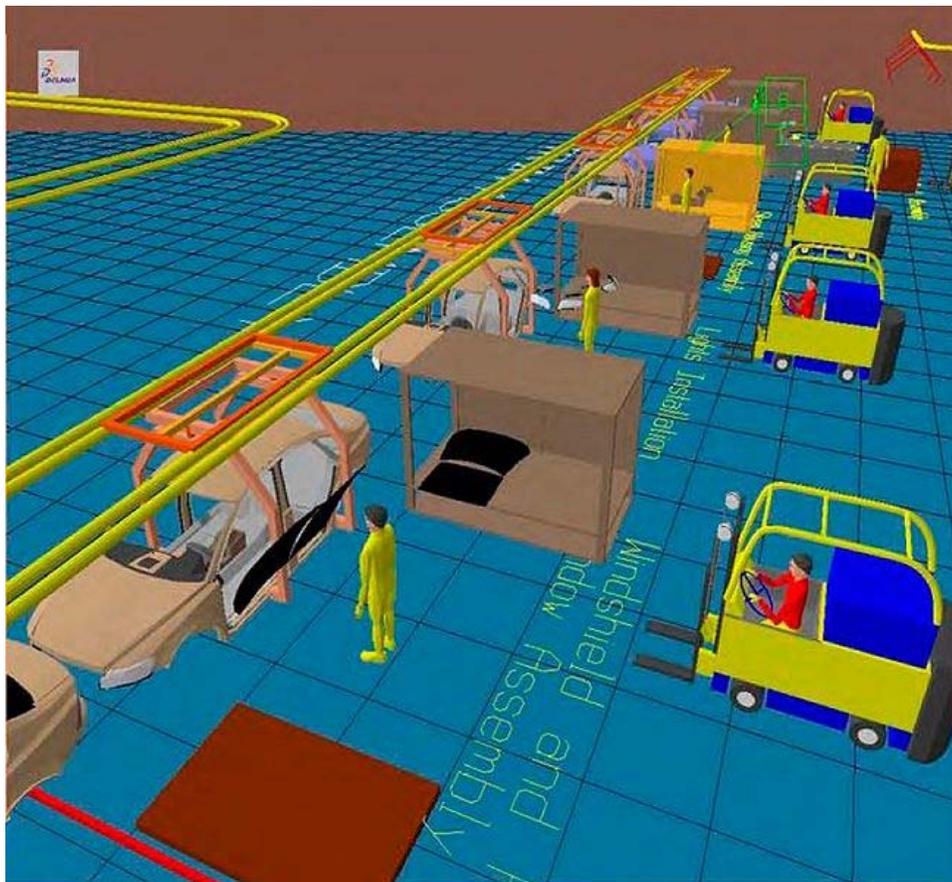
^** Like static elements dynamic elements, thermal elements and electrical elements

^ ^*** Steady, Static and Constant loads are the same. Transient loads include: quasi-static loads (slowly varying loads in which the effect of inertial is small enough to neglect) and dynamic loads (faster varying loads).

Chapter-2

Computer-integrated Manufacturing and Digital Read Out

Computer-integrated manufacturing



Manufacturing Systems Integration Program, NIST 2008.

Computer-integrated manufacturing (CIM) is the manufacturing approach of using computers to control the entire production process. This integration allows individual processes to exchange information with each other and initiate actions. Through the integration of computers, manufacturing can be faster and less error-prone, although the main advantage is the ability to create automated manufacturing processes. Typically CIM relies on closed-loop control processes, based on real-time input from sensors. It is also known as *flexible design and manufacturing*.

Overview

The term "computer-integrated manufacturing" is both a method of manufacturing and the name of a computer-automated system in which individual engineering, production, marketing, and support functions of a manufacturing enterprise are organized. In a CIM system functional areas such as design, analysis, planning, purchasing, cost accounting, inventory control, and distribution are linked through the computer with factory floor functions such as materials handling and management, providing direct control and monitoring of all the operations.

As a method of manufacturing, three components distinguish CIM from other manufacturing methodologies:

- Means for data storage, retrieval, manipulation and presentation;
- Mechanisms for sensing state and modifying processes;
- Algorithms for uniting the data processing component with the sensor/modification component.

CIM is an example of the implementation of information and communication technologies (ICTs) in manufacturing.

CIM implies that there are at least two computers exchanging information, e.g. the controller of an arm robot and a micro-controller of a CNC machine.

Some factors involved when considering a CIM implementation are the production volume, the experience of the company or personnel to make the integration, the level of the integration into the product itself and the integration of the production processes. CIM is most useful where a high level of ICT is used in the company or facility, such as CAD/CAM systems, the availability of process planning and its data.

History

The idea of "digital manufacturing" was prominent the 1980s, when computer-integrated manufacturing was developed and promoted by machine tool manufacturers and the Computer and Automated Systems Association and Society of Manufacturing Engineers (CASA/SME).

"CIM is the integration of total manufacturing enterprise by using integrated systems and data communication coupled with new managerial philosophies that improve organizational and personnel efficiency." ERHUM

Computer-integrated manufacturing topics

Key challenges

There are three major challenges to development of a smoothly operating computer-integrated manufacturing system:

- Integration of components from different suppliers: When different machines, such as CNC, conveyors and robots, are using different communications protocols. In the case of AGVs, even differing lengths of time for charging the batteries may cause problems.
- Data integrity: The higher the degree of automation, the more critical is the integrity of the data used to control the machines. While the CIM system saves on labor of operating the machines, it requires extra human labor in ensuring that there are proper safeguards for the data signals that are used to control the machines.
- Process control: Computers may be used to *assist* the human operators of the manufacturing facility, but there must always be a competent engineer on hand to handle circumstances which could not be foreseen by the designers of the control software.

Subsystems in computer-integrated manufacturing

A computer-integrated manufacturing system is not the same as a *"lights-out" factory*, which would run completely independent of human intervention, although it is a big step in that direction. Part of the system involves flexible manufacturing, where the factory can be quickly modified to produce different products, or where the volume of products can be changed quickly with the aid of computers. Some or all of the following subsystems may be found in a CIM operation:

Computer-aided techniques:

- CAD (computer-aided design)
- CAE (computer-aided engineering)
- CAM (computer-aided manufacturing)
- CAPP (computer-aided process planning)
- CAQ (computer-aided quality assurance)
- PPC (production planning and control)
- ERP (enterprise resource planning)
- A business system integrated by a common database.

Devices and equipment required:

- CNC, Computer numerical controlled machine tools
- DNC, Direct numerical control machine tools
- PLCs, Programmable logic controllers
- Robotics
- Computers
- Software
- Controllers
- Networks
- Interfacing
- Monitoring equipment

Technologies:

- FMS, (flexible manufacturing system)
- ASRS, automated storage and retrieval system
- AGV, automated guided vehicle
- Robotics
- Automated conveyance systems

Others:

- Lean manufacturing

CIMOSA

CIMOSA (Computer Integrated Manufacturing Open System Architecture), is a 1990s European proposal for an open system architecture for CIM developed by the AMICE Consortium as a series of ESPRIT projects. The goal of CIMOSA was "to help companies to manage change and integrate their facilities and operations to face world wide competition. It provides a consistent architectural framework for both enterprise modeling and enterprise integration as required in CIM environments".

CIMOSA provides a solution for business integration with four types of products:

- The CIMOSA Enterprise Modeling Framework, which provides a reference architecture for enterprise architecture
- CIMOSA IIS, a standard for physical and application integration.
- CIMOSA Systems Life Cycle, is a life cycle model for CIM development and deployment.
- Inputs to standardization, basics for international standard development.

CIMOSA according to Vernadat (1996), coined the term business process and introduced the process-based approach for integrated enterprise modeling based on a cross-boundaries approach, which opposed to traditional function or activity-based approaches.

With CIMOSA also the concept of an "Open System Architecture" (OSA) for CIM was introduced, which was designed to be vendor-independent, and constructed with standardised CIM modules. Here to the OSA is "described in terms of their function, information, resource, and organizational aspects. This should be designed with structured engineering methods and made operational in a modular and evolutionary architecture for operational use".

Application

There are multiple areas of usage:

- In mechanical engineering
- In electronic design automation (printed circuit board (PCB) and integrated circuit design data for manufacturing)

Digital read out



DRO providing a three axis display with pitch circle calculator, diameter/radius conversion, absolute and incremental toggle, and inch metric toggle

A **digital read out (DRO)** is a small computer (display unit) usually with an integrated keyboard and some means of numeric representation. It reads the signals generated by the linear encoder (or less frequently by rotary encoders) installed to several machine's axes used to keep track of workpiece position (milling and the like) or the tool's position (lathes and grinders). In the shop argot this complete system conformed by the computer and the encoders are referred to simply as **DRO**. Such a system is commonly fitted to most machines in today's shops: lathes, cylindrical grinders, milling machines, surface grinders, boring mills and other machine tools to allow the operator to work faster and with greater accuracy. The use of DROs are not limited to manual equipment as CNC equipment can usually be switched to manual operation and in this case a form of DRO is simulated on the Control Panel of a robotic machine.

Display unit (computer)

Several 7-segment displays, or an LCD screen on more expensive models display the current position of each machine axes. Such as X, Y, Z for a common milling machines and those plus U and W for highly sophisticated 5 axes vertical machining centers; x and z for a lathes or cylindrical grinders; and just z for a surface grinder.

Common Standard functions on a modern DRO

Modern DROs have a lot of functionality, giving you finger tip control over the most frequent shop operations. The following were taken from a popular brand User's manual:

1. Imperial (inch) and metric interchange.
2. The so called "1/2" function. It takes the current value of an axis and divides it by two. It is used to find the accurate center of a workpiece or an existing feature such as a hole.
3. Preset dimensions. The operator can type in the value of an axis. Used to locate a feature by using an edge finder.
4. Absolute or Incremental modes. The position of a feature given on a blue print is given by one of two methods:
 1. Absolute, which means the coordinate is relative to the part's absolute zero (usually one of its corners or its center).
 2. Relative, meaning the coordinate is referred to some other feature, usually the last one machined.
5. Bolt holes. The drilling or boring of several holes along an arch without using a rotary table.
6. Inclines, compute a cut across an incline.
7. Memory, Stores hundreds or thousands of points.
8. Scientific calculator. A full featured calculator has been integrated to some select models.

Linear encoders

All encoders have a scale that attaches to the moving part (the table, carriage, knee or quill) and a reader that attaches to the part that does not move. All are subject to damage from impact, so should be protected with some sort of metallic shield.

Glass scales

Made from strips of high quality glass with evenly etched marks just like the marks of a ruler, but very small (the smallest are just 5 μm apart). Two optical sensors (phototransistors or photodiodes) are placed very close to each other to make a linear encoder . When the machine axis moves, the dark marks move under the optical encoders triggering them in succession. If movement is from, for example, left to right, encoder A is triggered first and encoder B afterwards. So the computer can know that the scale

moved 5 μ m to the right. And, if encoder B triggers first and A does as follow the computer knows it was in the other direction.

Commercial models are enclosed in an aluminum "box" with a rubber protection on the side where the encoder slides. Used primarily where shielding from coolant and chips is necessary or where a resolution of 5 μ m (0.0002 in) or better is required (surface grinders).

Glass scales used to be too expensive for home use, but have recently fallen in price (late 2008) to where they are competitive with the other technologies.

Electronic scales

Instead of glass, a printed circuit on a stainless steel ruler is used to trigger at least two microelectronic hall effect sensors. Resolution is limited to 0.01mm (0.0005 in) but shielding from coolant and flying chip is not a requirement. These scales are very resistant to everyday shop contaminants and debris. Electronic scales are much cheaper than their glass counterparts.

Electronic scales are available with build in displays so they can be used independently.

Ball scales

The ball scales produced by Newall use an electromagnetic field to track ball bearings in a tube. They sell under the trade name Spherosyn and Microsyn. They work only with Newall DROs.

Rack and Gear scales

These models use a rack (toothed metal strip) that meshes with a gear that turns a rotary encoder. Claimed accuracy of .002 inches per foot, though users frequently report it is much more accurate, with no measurable deviation over several feet of travel. Debris getting between the gear and rack is a concern.

Quill DRO

Vertical Quill DRO



Vertical quill DRO

This is a speciality DRO system composed of a computer and an electronic scale in one small piece of equipment. Usually battery operated. Installed on the quill of a milling machine (whence its name). At this part of the machine coolant splash, flying chip and accidental shock are day-to-day events, therefore it is a very bad place for a glass scale used in conventional DROs. It also gives the operator a lot of comfort by being placed right in front of his eyes next to the controls that adjust the machine and it has not a single wire that can get trapped in the cluttered quill area. A very common setup is to have a regular DRO with glass scales on the milling machine's table and a separate

vertical quill DRO. This gives a resolution of 0.005mm to the table position and 0.01mm to the quill. Both exceed by far the expected "0.04mm accuracy of the milling process" (The Machinery's Handbook)

Horizontal quill DRO

It is very much like the vertical version, except in that it is designed to be installed in a horizontal position. The only difference is the orientation of the display and the buttons to be seen and operated horizontally. This device is not for standard shop equipment. It is used in research and calibration of other equipment.

Chapter-3

STEP-NC



STEP-NC interface on a CNC, showing product shape and color-coded tolerance state

STEP-NC is a machine tool control language that extends the ISO 10303 STEP standards with the machining model in ISO 14649, adding geometric dimension and tolerance data for inspection, and the STEP PDM model for integration into the wider enterprise. The combined result has been standardized as ISO 10303-238 (also known as AP238).

STEP-NC was designed to replace ISO 6983/RS274D G-codes with a modern, associative protocol that connects computer numerical controlled (CNC) process data to a product description of the part being machined.

A STEP-NC program can use the full range of geometric constructs from the STEP standard to communicate device-independent toolpaths to the CNC. It can provide CAM operational descriptions and STEP CAD geometry to the CNC so workpieces, stock, fixtures and cutting tool shapes can be visualized and analyzed in the context of the toolpaths. STEP GD&T information can also be added to enable quality measurement on the control, and CAM-independent volume removal features may be added to facilitate regeneration and modification of the toolpaths before or during machining for closed loop manufacturing.

Motivation



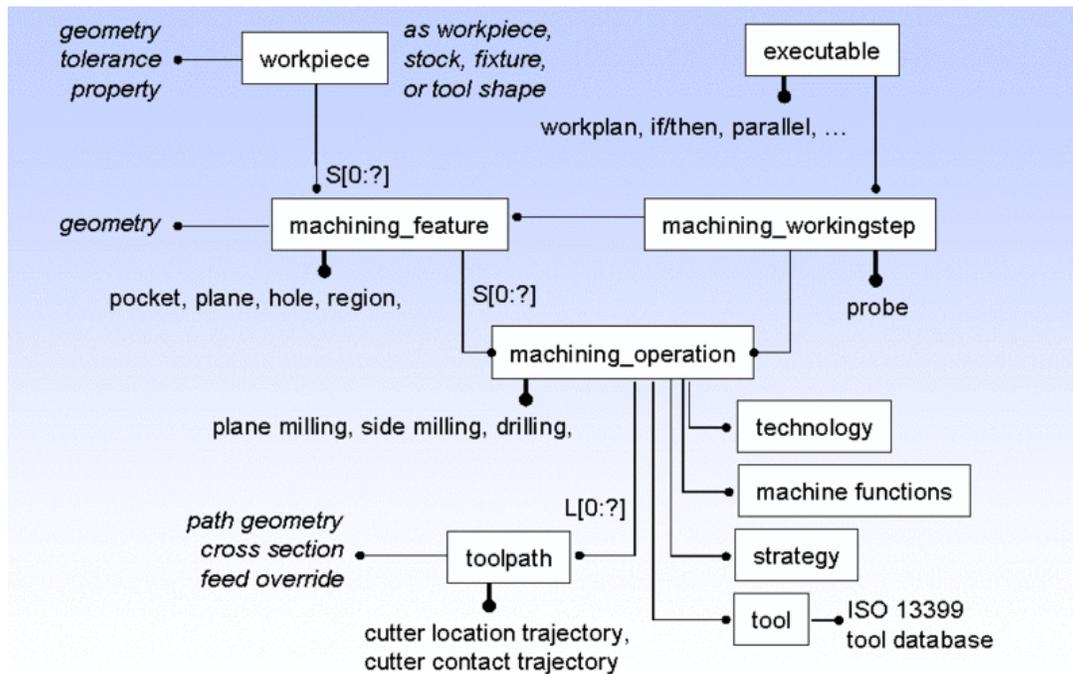
Impeller machined using STEP-NC

Input to a CNC in the ISO 6983/RS274D G-code control language is often machine-specific and limited to axis motion commands. The machine tool is given little or no information about the desired result of the machining.

STEP-NC allows more information about the machining process to be sent to the machine control and adds new information about the product being machined. This "Smart Data for Smart Machining" enables applications such as the following:

- Toolpath descriptions that are portable and independent of machine geometry.
- Visual process, to show toolpaths in context of the machine and workpiece, and eliminate drawings.
- On-Machine Simulation, to check for gouges, machine interference and other undesired behavior.
- Simplified Inspection, with linked tolerances, on-machine probes and inspection workplans tied to part tolerances.
- Feed and Speed Optimization, using tolerances, cross section information, sensor data.
- Associativity so feedback can be sent from manufacturing back to design.

Capabilities



Overview of STEP-NC process model

STEP-NC can communicate a complete machining process description to a machine tool control or between manufacturing software applications. The information handled by STEP-NC can be divided into the following general categories. The standard handles

technology-specific parameters for milling and turning, and extensions for other technologies under development.

- Product Description
 - Workpiece, PDM and Product Geometry
 - Manufacturing Features
 - Dimensions and Tolerances
 - Measures and Part Properties

- General Process Description
 - Project
 - Executable
 - Operation
 - Toolpath

- Technology-Specific Process Description
 - Operations and cutting tools for milling
 - Operations and cutting tools for turning
 - Operations and devices for inspection

STEP-NC can exchange the explicit toolpath descriptions in use today, and add part, stock, and fixture geometry, a description of the tools, geometric dimensions and tolerances, and PDM information. A STEP-NC file is difficult to edit by hand because it contains geometry descriptions but for large programs the file size can be smaller because STEP-NC uses a compressed XML format instead of ASCII codes.

History

STEP-NC is not the first attempt at providing better quality information to a CNC. The EIA 494 Basic Control Language (BCL) defined a control language that was portable and had toolpaths independent of machine geometry, but did not contain any of the other product model information found in STEP-NC.

The core of STEP-NC is the ISO 14649 model for CNC control developed by European ESPRIT and IMS STEP-NC projects begun in 1999. These were led by Siemens with contributions from the University of Aachen and the University of Stuttgart in Germany, Komatsu and FANUC in Japan, Heidenhain in Switzerland, and the Pohang University of Science and Technology in Korea. Models for the control of CNC milling and turning machines were published in 2005, and draft models exist for EDM and contour cutting.

Integration of the CNC model into STEP to produce ISO 10303-238 was done in the United States, under the NIST ATP Model Driven Intelligent Control of Manufacturing project, led by STEP Tools, Inc. with an industrial review board (IRB) consisting of Fortune 500 companies, CAD and CAM software developers, machine tool manufacturers, job shops and industry experts. STEP-NC AP238 was published in 2007.

In 2005 the OMAC STEP-NC Working Group hosted an AP238 testing forum in Orlando to demonstrate 5-axis parts machined using AP238 CC1 machine independent toolpaths. Four CAD/CAM systems produced AP238 machining programs for milling a 5-axis test part (an NAS 979 circle/diamond/square with an inverted NAS 979 cone test in the center). Each run on a pair of CNCs configured for completely different machine geometries (AB tool tilt vs. BC table tilt). In addition, Boeing cut parts on a variety of machines at their Tulsa facility and a machine at NIST in Gaithersburg.

In June 2006, a live 5-axis STEP-NC machining demonstration was hosted by Airbus at the Université Paul Sabatier Laboratoire de Génie mécanique in Toulouse. Further machining and measurement demonstrations were conducted in Ibusuki Japan in 2007.

On March 10–12, 2008, the STEP Manufacturing team (ISO TC184 SC4 WG3 T24) met in Sandviken and Stockholm, Sweden to demonstrate use of STEP-NC for feed and speed optimization, high-speed machining, tolerance-driven tool compensation and traceability. The participants in the demonstrations included Airbus/Univ. Bordeaux, Boeing, Eurostep, KTH Royal Institute of Technology, NIST, Sandvik Coromant, Scania, STEP Tools, and Univ. of Vigo.

On October 1–2, 2008, the STEP Manufacturing team met at the Connecticut Center for Advanced Technology, in Hartford, Connecticut to demonstrate closed-loop machining, feed optimization, and measurement using STEP-NC. The highlight of the meeting was the live 5-axis machining of a titanium impeller. Participants in the machining demonstration and other activities included Boeing, Connecticut Center for Advanced Technology, Concepts NRec, DMG, KTH Royal Institute of Technology, Mitutoyo, NIST, Sandvik Coromant, Scania, Siemens, and STEP Tools.

These participants and others continue to hold STEP-NC international implementation and testing events on a roughly six month cycle. The demonstrations in 2009 focused on machining a Mold part at multiple sites from the same AP238 data including one part machined on a FANUC-developed STEP-NC control. At a meeting in Seattle the parts were then measured for accuracy using a CMM probe and a laser scanner.

In the first half of 2010, the testing activity focused on tool wear management and machining a part in multiple setups with multiple alternate machining plans for 3, 4 and 5-axis machining. The new test part was a gear box that must be machined on all six sides. The tool wear and consequent machine loads were predicted from the STEP-NC data and verified using a dynamometer. In the second half of 2010, the testing forum applied STEP-NC to setup compensation with on-machine measurement of part and fixture datums using a FaroArm portable measurement device.

Future Work



STEP-NC plasma cutting

Work continues within the ISO standard committees to extend STEP-NC to new technologies and to incorporate refinements discovered during use. Process models for new technologies are usually produced by the ISO TC184/SC1/WG7 committee. Models for Wire & Sink EDM and contour cutting of wood or stone are under investigation.

Work on extending and integrating STEP-NC with the manufacturing enterprise takes place in the ISO TC184/SC4/WG3/T24 STEP Manufacturing Team. This group also works on extensions and refinements discovered during testing. A series of traceability extensions have been proposed for linking STEP-NC machining programs with sensor feedback and machine state information during execution.

The National Shipbuilding Research Program (NSRP) has also hosted work to implement a prototype that connects a shipyard design system to a plate cutting using STEP-NC. This work involved extending STEP-NC to steel plate cutting and marking using lasers and plasma torches.

Chapter-4

Numerical Control



A CNC Turning Center.



Siemens CNC panel.

Numerical control (NC) refers to the automation of machine tools that are operated by abstractly programmed commands encoded on a storage medium, as opposed to manually controlled via handwheels or levers, or mechanically automated via cams alone. The first NC machines were built in the 1940s and 1950s, based on existing tools that were modified with motors that moved the controls to follow points fed into the system on punched tape. These early servomechanisms were rapidly augmented with analog and digital computers, creating the modern **computer numerical control (CNC)** machine tools that have revolutionized the manufacturing process.

In modern CNC systems, end-to-end component design is highly automated using computer-aided design (CAD) and computer-aided manufacturing (CAM) programs. The

programs produce a computer file that is interpreted to extract the commands needed to operate a particular machine via a postprocessor, and then loaded into the CNC machines for production. Since any particular component might require the use of a number of different tools-drills, saws, etc., modern machines often combine multiple tools into a single "cell". In other cases, a number of different machines are used with an external controller and human or robotic operators that move the component from machine to machine. In either case, the complex series of steps needed to produce any part is highly automated and produces a part that closely matches the original CAD design.

History

Earlier forms of automation

Cams

The automation of machine tool control began in the 19th century with cams that "played" a machine tool in the way that cams had long been playing musical boxes or operating elaborate cuckoo clocks. Thomas Blanchard built his gun-stock-copying lathes (1820s-30s), and the work of people such as Christopher Miner Spencer developed the turret lathe into the screw machine (1870s). Cam-based automation had already reached a highly advanced state by World War I (1910s).

However, automation via cams is fundamentally different from numerical control because it cannot be abstractly programmed. Cams can encode information, but getting the information from the abstract level of an engineering drawing into the cam is a manual process that requires sculpting and/or machining and filing.

Various forms of abstractly programmable control had existed during the 19th century: those of the Jacquard loom, player pianos, and mechanical computers pioneered by Charles Babbage and others. These developments had the potential for convergence with the automation of machine tool control starting in that century, but the convergence did not happen until many decades later.

Tracer control

The application of hydraulics to cam-based automation resulted in tracing machines that used a stylus to trace a template, such as the enormous Pratt & Whitney "Keller Machine", which could copy templates several feet across. Another approach was "record and playback", pioneered at General Motors (GM) in the 1950s, which used a storage system to record the movements of a human machinist, and then play them back on demand. Analogous systems are common even today, notably the "teaching lathe" which gives new machinists a hands-on feel for the process. None of these were numerically programmable, however, and required a master machinist at some point in the process, because the "programming" was physical rather than numerical.

Servos and selsyns

One barrier to complete automation was the required tolerances of the machining process, which are routinely on the order of thousandths of an inch. Although connecting some sort of control to a storage device like punched cards was easy, ensuring that the controls were moved to the correct position with the required accuracy was another issue. The movement of the tool resulted in varying forces on the controls that would mean a linear input would not result in linear tool motion. The key development in this area was the introduction of the servomechanism, which produced highly accurate measurement information. Attaching two servos together produced a selsyn, where a remote servo's motions were accurately matched by another. Using a variety of mechanical or electrical systems, the output of the selsyns could be read to ensure proper movement had occurred (in other words, forming a closed-loop control system).

The first serious suggestion that selsyns could be used for machining control was made by Ernst F. W. Alexanderson, a Swedish immigrant to the U.S. working at General Electric (GE). Alexanderson had worked on the problem of torque amplification that allowed the small output of a mechanical computer to drive very large motors, which GE used as part of a larger gun laying system for US Navy ships. Like machining, gun laying requires very high accuracies, much less than a degree, and the forces during the motion of the gun turrets was non-linear. In November 1931 Alexanderson suggested to the Industrial Engineering Department that the same systems could be used to drive the inputs of machine tools, allowing it to follow the outline of a template without the strong physical contact needed by existing tools like the Keller Machine. He stated that it was a "matter of straight engineering development". However, the concept was ahead of its time from a business development perspective, and GE did not take the matter seriously until years later, when others had pioneered the field.

Parsons and the invention of NC

The birth of NC is generally credited to John T. Parsons, a machinist and salesman at his father's machining company, Parsons Corp.

In 1942 he was told that helicopters were going to be the "next big thing" by the former head of Ford Trimotor production, Bill Stout. He called Sikorsky Aircraft to inquire about possible work, and soon got a contract to build the wooden stringers in the rotor blades. After setting up production at a disused furniture factory and ramping up production, one of the blades failed and it was traced to the spar. As at least some of the problem appeared to stem from spot welding a metal collar on the stringer to the metal spar, so Parsons suggested a new method of attaching the stringers to the spar using adhesives, never before tried on an aircraft design.

That development led Parsons to consider the possibility of using stamped metal stringers instead of wood, which would be much stronger and easier to make. The stringers for the rotors were built from a design provided by Sikorsky, which was sent to Parsons as a series of 17 points defining the outline. Parsons then had to "fill in" the dots with a french

curve to generate an outline they could use as a template to build the jigs for the wooden stringers. Making a metal cutting tool able to cut that particular shape proved to be difficult. Parsons went to Wright Field to see Frank Stulen, the head of the Propeller Lab Rotary Ring Branch. During their conversation, Stulen concluded that Parsons didn't really know what he was talking about. Parsons realized this, and hired Stulen on the spot. Stulen started work on 1 April 1946 and hired three new engineers to join him.

Stulen's brother worked at Curtis Wright Propeller, and mentioned that they were using punched card calculators for engineering calculations. Stulen decided to adopt the idea to run stress calculations on the rotors, the first detailed automated calculations on helicopter rotors. When Parsons saw what Stulen was doing with the punched card machines, he asked Stulen if they could be used to generate an outline with 200 points instead of the 17 they were given, and offset each point by the radius of a mill cutting tool. If you cut at each of those points, it would produce a relatively accurate cutout of the stringer even in hard steel, and it could easily be filed down to a smooth shape. The resulting tool would be useful as a template for stamping metal stringers. Stullen had no problem making such a program, and used it to produce large tables of numbers that would be taken onto the machine floor. Here, one operator read the numbers off the charts to two other operators, one on each of the X- and Y- axes, and they would move the cutting head to that point and make a cut. This was called the "by-the-numbers method".

At that point Parsons conceived of a fully automated tool. With enough points on the outline, no manual working would be needed, but with manual operation, the time saved by having the part more closely match the outline was offset by the time needed to move the controls. If the machine's inputs were attached directly to the card reader, this delay, and any associated manual errors, would be removed and the number of points could be dramatically increased. Such a machine could repeatedly punch out perfectly accurate templates on command. But at the time Parsons had no funds to develop his ideas.

When one of Parsons's salesmen was on a visit to Wright Field, he was told of the problems the newly-formed US Air Force was having with new jet designs. He asked if Parsons had anything to help them. Parsons showed Lockheed their idea of an automated mill, but they were uninterested. They decided to use 5-axis template copiers to produce the stringers, cutting from a metal template, and had already ordered the expensive cutting machine. But as Parsons noted:

Now just picture the situation for a minute. Lockheed had contracted to design a machine to make these wings. This machine had five axes of cutter movement, and each of these was tracer controlled using a template. Nobody was using my method of making templates, so just imagine what chance they were going to have of making an accurate airfoil shape with inaccurate templates.

Parsons worries soon came true, and Lockheed's protests that they could fix the problem eventually rang hollow. In 1949 the Air Force arranged funding for Parsons to build his machines on his own. Early work with Snyder Machine & Tool Corp proved the system

of directly driving the controls from motors failed to give the accuracy needed to set the machine for a perfectly smooth cut. Since the mechanical controls did not respond in a linear fashion, you couldn't simply drive it with a given amount of power, because the differing forces meant the same amount of power would not always produce the same amount of motion in the controls. No matter how many points you included, the outline would still be rough.

Enter MIT

This was not an impossible problem to solve, but would require some sort of feedback system, like a selsyn, to directly measure how far the controls had actually turned. Faced with the daunting task of building such a system, in the spring of 1949 Parsons turned to Gordon S. Brown's Servomechanisms Laboratory at MIT, which was a world leader in mechanical computing and feedback systems. During the war the Lab had built a number of complex motor-driven devices like the motorized gun turret systems for the Boeing B-29 Superfortress and the automatic tracking system for the SCR-584 radar. They were naturally suited to technological transfer into a prototype of Parsons's automated "by-the-numbers" machine.

The MIT team was led by William Pease assisted by James McDonough. They quickly concluded that Parsons's design could be greatly improved; if the machine did not simply cut *at* points A and B, but instead moved smoothly *between* the points, then not only would it make a perfectly smooth cut, but could do so with many fewer points - the mill could cut lines directly instead of having to define a large number of cutting points to "simulate" it. A three-way agreement was arranged between Parsons, MIT, and the Air Force, and the project officially ran from July 1949 to June 1950. The contract called for the construction of two "Card-a-matic Milling Machine"s, a prototype and a production system. Both to be handed to Parsons for attachment to one of their mills in order to develop a deliverable system for cutting stringers.

Instead, in 1950 MIT bought a surplus Cincinnati Milling Machine Company "Hydro-Tel" mill of their own and arranged a new contract directly with the Air Force that froze Parsons out of further development. Parsons would later comment that he "never dreamed that anybody as reputable as MIT would deliberately go ahead and take over my project." In spite of the development being handed to MIT, Parsons filed for a patent on "Motor Controlled Apparatus for Positioning Machine Tool" on 5 May 1952, sparking a filing by MIT for a "Numerical Control Servo-System" on 14 August 1952. Parsons received US Patent 2,820,187 on 14 January 1958, and the company sold an exclusive license to Bendix. IBM, Fujitsu and General Electric all took sub-licenses after having already started development of their own devices.

MIT's machine

MIT fit gears to the various handwheel inputs and drove them with roller chains connected to motors, one for each of the machine's three axes (X, Y, and Z). The associated controller consisted of five refrigerator-sized cabinets that, together, were

almost as large as the mill they were connected to. Three of the cabinets contained the motor controllers, one controller for each motor, the other two the digital reading system.

Unlike Parsons's original punched card design, the MIT design used standard 7-track punch tape for input. Three of the tracks were used to control the different axes of the machine, while the other four encoded various control information. The tape was read in a cabinet that also housed six relay-based hardware registers, two for each axis. With every read operation the previously read point was copied into the "starting point" register, and the newly read one into the "ending point". The tape was read continually and the number in the register increased until a "stop" instruction was encountered, four holes in a line.

The final cabinet held a clock that sent pulses through the registers, compared them, and generated output pulses that interpolated between the points. For instance, if the points were far apart the output would have pulses with every clock cycle, whereas closely spaced points would only generate pulses after multiple clock cycles. The pulses are sent into a summing register in the motor controllers, counting up by the number of pulses every time they were received. The summing registers were connected to a digital to analog converter that increased power to the motors as the count in the registers increased.

The registers were decremented by encoders attached to the motors and the mill itself, which would reduce the count by one for every one degree of rotation. Once the second point was reached the pulses from the clock would stop, and the motors would eventually drive the mill to the encoded position. Each 1 degree rotation of the controls produced a 0.0005 inch movement of the cutting head. The programmer could control the speed of the cut by selecting points that were closer together for slow movements, or further apart for rapid ones.

The system was publicly demonstrated in September 1952, appearing in that month's *Scientific American*. MIT's system was an outstanding success by any technical measure, quickly making any complex cut with extremely high accuracy that could not easily be duplicated by hand. However, the system was terribly complex, including 250 vacuum tubes, 175 relays and numerous moving parts, reducing its reliability in a production environment. It was also very expensive, the total bill presented to the Air Force was \$360,000.14, \$2,641,727.63 in 2005 dollars. Between 1952 and 1956 the system was used to mill a number of one-off designs for various aviation firms, in order to study their potential economic impact.

Proliferation of NC

The Air Force funding for the project ran out in 1953, but development was picked up by the Giddings and Lewis Machine Tool Co. In 1955 many of the MIT team left to form Concord Controls, a commercial NC company with Giddings' backing, producing the Numericord controller. Numericord was similar to the MIT design, but replaced the punch tape with a magnetic tape reader that General Electric was working on. The tape

contained a number of signals of different phases, which directly encoded the angle of the various controls. The tape was played at a constant speed in the controller, which set its half of the selsyn to the encoded angles while the remote side was attached to the machine controls. Designs were still encoded on paper tape, but the tapes were transferred to a reader/writer that converted them into magnetic form. The magtapes could then be used on any of the machines on the floor, where the controllers were greatly reduced in complexity. Developed to produce highly accurate dies for an aircraft skinning press, the Numericord "NC5" went into operation at G&L's plant at Fond du Lac, WI in 1955.

Monarch Machine Tool also developed an numerical controlled lathe, starting in 1952. They demonstrated their machine at the 1955 Chicago Machine Tool Show, along with a number of other vendors with punched card or paper tape machines that were either fully developed or in prototype form. These included Kearney & Trecker's Milwaukee-Matic II that could change its cutting tool under numerical control, a common feature on modern machines.

A Boeing report noted that "numerical control has proved it can reduce costs, reduce lead times, improve quality, reduce tooling and increase productivity." In spite of these developments, and glowing reviews from the few users, uptake of NC was relatively slow. As Parsons later noted:

The NC concept was so strange to manufacturers, and so slow to catch on, that the US Army itself finally had to build 120 NC machines and lease them to various manufacturers to begin popularizing its use.

In 1958 MIT published its report on the economics of NC. They concluded that the tools were competitive with human operators, but simply moved the time from the machining to the creation of the tapes. In *Forces of Production*, Noble claims that this was the whole point as far as the Air Force was concerned; moving the process off of the highly unionized factory floor and into the un-unionized white collar design office. The cultural context of the early 1950s, a second Red Scare with a widespread fear of a bomber gap and of domestic subversion, sheds light on this interpretation. It was strongly feared that the West would lose the defense production race to the Communists, and that syndicalist power was a path toward losing, either by "getting too soft" (less output, greater unit expense) or even by Communist sympathy and subversion within unions (arising from their common theme of empowering the working class).

CNC arrives

Many of the commands for the experimental parts were programmed "by hand" to produce the punch tapes that were used as input. During the development of Whirlwind, MIT's real-time computer, John Runyon coded a number of subroutines to produce these tapes under computer control. Users could enter a list of points and speeds, and the program would generate the punch tape. In one instance, this process reduced the time required to produce the instruction list and mill the part from 8 hours to 15 minutes. This

led to a proposal to the Air Force to produce a generalized "programming" language for numerical control, which was accepted in June 1956.

Starting in September, Ross and Pople outlined a language for machine control that was based on points and lines, developing this over several years into the APT programming language. In 1957 the Aircraft Industries Association (AIA) and Air Material Command at Wright-Patterson Air Force Base joined with MIT to standardize this work and produce a fully computer-controlled NC system. On 25 February 1959 the combined team held a press conference showing the results, including a 3D machined aluminum ash tray that was handed out in the press kit.

Meanwhile, Patrick Hanratty was making similar developments at GE as part of their partnership with G&L on the Numericord. His language, PRONTO, beat APT into commercial use when it was released in 1958. Hanratty then went on to develop MICR magnetic ink characters that were used in cheque processing, before moving to General Motors to work on the groundbreaking DAC-1 CAD system.

APT was soon extended to include "real" curves in 2D-APT-II. With its release, MIT reduced its focus on CNC as it moved into CAD experiments. APT development was picked up with the AIA in San Diego, and in 1962, by Illinois Institute of Technology Research. Work on making APT an international standard started in 1963 under USASI X3.4.7, but many manufacturers of CNC machines had their own one-off additions (like PRONTO), so standardization was not completed until 1968, when there were 25 optional add-ins to the basic system.

Just as APT was being released in the early 1960s, a second generation of lower-cost transistorized computers was hitting the market that were able to process much larger volumes of information in production settings. This reduced the cost of implementing a NC system and by the mid 1960s, APT runs accounted for a third of all computer time at large aviation firms.

CAD meets CNC

While the Servomechanisms Lab was in the process of developing their first mill, in 1953, MIT's Mechanical Engineering Department dropped the requirement that undergraduates take courses in drawing. The instructors formerly teaching these programs were merged into the Design Division, where an informal discussion of computerized design started. Meanwhile the Electronic Systems Laboratory, the newly rechristened Servomechanisms Laboratory, had been discussing whether or not design would ever start with paper diagrams in the future.

In January 1959, an informal meeting was held involving individuals from both the Electronic Systems Laboratory and the Mechanical Engineering Department's Design Division. Formal meetings followed in April and May, which resulted in the "Computer-Aided Design Project". In December 1959, the Air Force issued a one year contract to ESL for \$223,000 to fund the Project, including \$20,800 earmarked for 104 hours of

computer time at \$200 per hour. This proved to be far too little for the ambitious program they had in mind, although their engineering calculation system, AED, was released in March 1965.

In 1959, General Motors started an experimental project to digitize, store and print the many design sketches being generated in the various GM design departments. When the basic concept demonstrated that it could work, they started the DAC-1 project with IBM to develop a production version. One part of the DAC project was the direct conversion of paper diagrams into 3D models, which were then converted into APT commands and cut on milling machines. In November 1963 a trunk lid design moved from 2D paper sketch to 3D clay prototype for the first time. With the exception of the initial sketch, the design-to-production loop had been closed.

Meanwhile, MIT's offsite Lincoln Labs was building computers to test new transistorized designs. The ultimate goal was essentially a transistorized Whirlwind known as TX-2, but in order to test various circuit designs a smaller version known as TX-0 was built first. When construction of TX-2 started, time in TX-0 freed up and this led to a number of experiments involving interactive input and use of the machine's CRT display for graphics. Further development of these concepts led to Ivan Sutherland's groundbreaking Sketchpad program on the TX-2.

Sutherland moved to the University of Utah after his Sketchpad work, but it inspired other MIT graduates to attempt the first true CAD system. It was Electronic Drafting Machine (EDM), sold to Control Data and known as "Digigraphics", that Lockheed used to build production parts for the C-5 Galaxy, the first example of an end-to-end CAD/CNC production system.

By 1970 there were a wide variety of CAD firms including Intergraph, Applicon, Computervision, Auto-trol Technology, UGS Corp. and others, as well as large vendors like CDC and IBM.

Proliferation of CNC

The price of computer cycles fell drastically during the 1960s with the widespread introduction of useful minicomputers. Eventually it became less expensive to handle the motor control and feedback with a computer program than it was with dedicated servo systems. Small computers were dedicated to a single mill, placing the entire process in a small box. PDP-8's and Data General Nova computers were common in these roles. The introduction of the microprocessor in the 1970s further reduced the cost of implementation, and today almost all CNC machines use some form of microprocessor to handle all operations.

The introduction of lower-cost CNC machines radically changed the manufacturing industry. Curves are as easy to cut as straight lines, complex 3-D structures are relatively easy to produce, and the number of machining steps that required human action have been dramatically reduced. With the increased automation of manufacturing processes

with CNC machining, considerable improvements in consistency and quality have been achieved with no strain on the operator. CNC automation reduced the frequency of errors and provided CNC operators with time to perform additional tasks. CNC automation also allows for more flexibility in the way parts are held in the manufacturing process and the time required to change the machine to produce different components.

During the early 1970s the Western economies were mired in slow economic growth and rising employment costs, and NC machines started to become more attractive. The major U.S. vendors were slow to respond to the demand for machines suitable for lower-cost NC systems, and into this void stepped the Germans. In 1979, sales of German machines surpassed the U.S. designs for the first time. This cycle quickly repeated itself, and by 1980 Japan had taken a leadership position, U.S. sales dropping all the time. Once sitting in the #1 position in terms of sales on a top-ten chart consisting entirely of U.S. companies in 1971, by 1987 Cincinnati Milacron was in 8th place on a chart heavily dominated by Japanese firms.

Many researchers have commented that the U.S. focus on high-end applications left them in an uncompetitive situation when the economic downturn in the early 1970s led to greatly increased demand for low-cost NC systems. Unlike the U.S. companies, who had focused on the highly profitable aerospace market, German and Japanese manufacturers targeted lower-profit segments from the start and were able to enter the low-cost markets much more easily.

As computing and networking evolved, so did direct numerical control (DNC). Its long-term coexistence with less networked variants of NC and CNC is explained by the fact that individual firms tend to stick with whatever is profitable, and their time and money for trying out alternatives is limited. This explains why machine tool models and tape storage media persist in grandfathered fashion even as the state of the art advances.

DIY, Hobby, and Personal CNC

Recent developments in small scale CNC have been enabled, in large part, by the Enhanced Machine Controller project from the National Institute of Standards and Technology (NIST), an agency of the US Government's Department of Commerce. EMC is a public domain program operating under the Linux operating system and working on PC based hardware. After the NIST project ended, development continued, leading to EMC2 which is licensed under the GNU General Public License and Lesser GNU General Public License (GPL and LGPL). Derivations of the original EMC software have also led to several proprietary PC based programs notably TurboCNC, and Mach3, as well as embedded systems based on proprietary hardware. The availability of these PC based control programs has led to the development of DIY CNC, allowing hobbyists to build their own using open source hardware designs. The same basic architecture has allowed manufacturers, such as Sherline and Taig, to produce turnkey lightweight desktop milling machines for hobbyists.

The easy availability of PC based software and support information of Mach3, written by Art Fenerty, lets anyone with some time and technical expertise make complex parts for home and prototype use. Fenerty is considered a principal founder of Windows-based PC CNC machining.

Eventually, the homebrew architecture was fully commercialized and used to create larger machinery suitable for commercial and industrial applications. This class of equipment has been referred to as Personal CNC. Parallel to the evolution of personal computers, Personal CNC has its roots in EMC and PC based control, but has evolved to the point where it can replace larger conventional equipment in many instances. As with the Personal Computer, Personal CNC is characterized by equipment whose size, capabilities, and original sales price make it useful for individuals, and which is intended to be operated directly by an end user, often without professional training in CNC technology.

Today

Although modern data storage techniques have moved on from punch tape in almost every other role, tapes are still relatively common in CNC systems. Several reasons explain this. One is easy backward compatibility of existing programs. Companies were spared the trouble of re-writing existing tapes into a new format. Another is the principle, mentioned earlier, that individual firms tend to stick with whatever is profitable, and their time and money for trying out alternatives is limited. A small firm that has found a profitable niche may keep older equipment in service for years because "if it ain't broke [profitability-wise], don't fix it." Competition places natural limits on that approach, as some amount of innovation and continuous improvement eventually becomes necessary, lest competitors be the ones who find the way to the "better mousetrap".

One change that was implemented fairly widely was the switch from paper to mylar tapes, which are much more mechanically robust. Floppy disks, USB flash drives and local area networking have replaced the tapes to some degree, especially in larger environments that are highly integrated.

The proliferation of CNC led to the need for new CNC standards that were not encumbered by licensing or particular design concepts, like APT. A number of different "standards" proliferated for a time, often based around vector graphics markup languages supported by plotters. One such standard has since become very common, the "G-code" that was originally used on Gerber Scientific plotters and then adapted for CNC use. The file format became so widely used that it has been embodied in an EIA standard. In turn, while G-code is the predominant language used by CNC machines today, there is a push to supplant it with STEP-NC, a system that was deliberately designed for CNC, rather than grown from an existing plotter standard.

While G-code is the most common method of programming, some machine-tool/control manufacturers also have invented their own proprietary "conversational" methods of programming, trying to make it easier to program simple parts and make set-up and

modifications at the machine easier (such as Mazak's Mazatrol and Hurco). These have met with varying success.

A more recent advancement in CNC interpreters is support of logical commands, known as parametric programming (also known as macro programming). Parametric programs include both device commands as well as a control language similar to BASIC. The programmer can make if/then/else statements, loops, subprogram calls, perform various arithmetic, and manipulate variables to create a large degree of freedom within one program. An entire product line of different sizes can be programmed using logic and simple math to create and scale an entire range of parts, or create a stock part that can be scaled to any size a customer demands.

Since about 2006, the idea has been suggested and pursued to foster the convergence with CNC and DNC of several trends elsewhere in the world of information technology that have not yet much affected CNC and DNC. One of these trends is the combination of greater data collection (more sensors), greater and more automated data exchange (via building new, open industry-standard XML schemas), and data mining to yield a new level of business intelligence and workflow automation in manufacturing. Another of these trends is the emergence of widely published APIs together with the aforementioned open data standards to encourage an ecosystem of user-generated apps and mashups, which can be both open and commercial—in other words, taking the new IT culture of app marketplaces that began in web development and smartphone app development and spreading it to CNC, DNC, and the other factory automation systems that are networked with the CNC/DNC. MTConnect is a leading effort to bring these ideas into successful implementation.

Description

Modern CNC mills differ little in concept from the original model built at MIT in 1952. Mills typically consist of a table that moves in the X and Y axes, and a tool spindle that moves in the Z (depth). The position of the tool is driven by motors through a series of step-down gears in order to provide highly accurate movements, or in modern designs, direct-drive stepper motors. Closed-loop control is not mandatory today, as open-loop control works as long as the forces are kept small enough.

As the controller hardware evolved, the mills themselves also evolved. One change has been to enclose the entire mechanism in a large box as a safety measure, often with additional safety interlocks to ensure the operator is far enough from the working piece for safe operation. Most new CNC systems built today are completely electronically controlled.

CNC-like systems are now used for any process that can be described as a series of movements and operations. These include laser cutting, welding, friction stir welding, ultrasonic welding, flame and plasma cutting, bending, spinning, pinning, gluing, fabric cutting, sewing, tape and fiber placement, routing, picking and placing (PnP), and sawing.

Tools with CNC variants

- Drills
- EDMs
- Lathes
- Milling machines
- Wood routers
- Sheet metal works (Turret Punch)
- Wire bending machines
- Hot-wire foam cutters
- Plasma cuttings
- Water jet cutters
- Laser cutting
- Oxy-fuel
- Surface grinders
- Cylindrical grinders
- 3D Printing
- Induction hardening machines

Tool / Machine crashing

In CNC, a *crash* is causing the automated machinery to move in a way that is harmful to the machine or tools, sometimes resulting in bending or breakage of cutting tools, accessory clamps, vises, and fixtures, or causing damage to the machine itself by bending guide rails, breaking drive screws, or causing structural components to crack under strain and break apart.

Many CNC tools have no inherent sense of the absolute position of the table or tools when turned on. They must be manually "homed" or "zeroed" to have any reference to work from, and these limits are just for figuring out the location of the part to work with it, and aren't really any sort of hard motion limit on the mechanism. It is often possible to drive the machine outside the physical bounds of its drive mechanism, resulting in a collision with itself or damage to the drive mechanism.

Many CNC tools also don't know anything about their working environment. They often lack any form of sensory capability to detect problems with the machining process, and will not abort if something goes wrong. They blindly follow the machining code provided and it is up to an operator to detect if a crash is either occurring or about to occur, and for the operator to manually abort the cutting process.

If the drive system is weaker than the machine structural integrity, then the drive system simply pushes against the obstruction and the drive motors "slip in place". The machine tool may not detect the collision or the slipping, so for example the tool should now be at 210mm on the X axis but is in fact at 32mm where it hit the obstruction and kept slipping. All of the next tool motions will be off by -178mm on the X axis, and all future

motions are now invalid, which may result in further collisions with clamps, vises, or the machine itself.

Collision detection and avoidance is possible, through the use of absolute position sensors (optical encoder strips or disks) to verify that motion occurred, or torque sensors or power-draw sensors on the drive system to detect abnormal strain when the machine should just be moving and not cutting, but these are not a common component of most CNC tools.

Instead, most CNC tools simply rely on the assumed accuracy of stepper motors that rotate a specific number of degrees in response to magnetic field changes. It is often assumed the stepper is perfectly accurate and never mis-steps, so tool position monitoring simply involves counting the number of pulses sent to the stepper over time. An alternate means of stepper position monitoring is usually not available, so crash or slip detection is not possible.

Chapter-5

Material Requirements Planning and Direct Numerical Control

Material requirements planning

Material requirements planning (MRP) is a production planning and inventory control system used to manage manufacturing processes. Most MRP systems are software-based, while it is possible to conduct MRP by hand as well.

An MRP system is intended to simultaneously meet three objectives:

- Ensure materials are available for production and products are available for delivery to customers.
- Maintain the lowest possible level of inventory.
- Plan manufacturing activities, delivery schedules and purchasing activities.

History

Prior to MRP and before computers dominated the industry, reorder-point/reorder-quantity (ROP/ROQ) type methods like EOQ had been used in manufacturing and inventory management. In the 1960s, Joseph Orlicky studied the TOYOTA Manufacturing Program and developed Material Requirements Planning (MRP), and Oliver Wight and George Plossl then developed MRP into manufacturing resource planning (MRP II). Orlicky's book is entitled *The New Way of Life in Production and Inventory Management* (1975). By 1975, MRP was implemented in 150 companies. This number had grown to about 8,000 by 1981. In the 1980s, Joe Orlicky's MRP evolved into Oliver Wight's manufacturing resource planning (MRP II) which brings master scheduling, rough-cut capacity planning, capacity requirements planning and other concepts to classical MRP. By 1989, about one third of the software industry was MRP II software sold to American industry (\$1.2 billion worth of software).

The scope of MRP in manufacturing

The basic function of MRP system includes inventory control, bill of material processing and elementary scheduling. MRP helps organizations to maintain low inventory levels. It is used to plan manufacturing, purchasing and delivering activities.

"Manufacturing organizations, whatever their products, face the same daily practical problem - that customers want products to be available in a shorter time than it takes to make them. This means that some level of planning is required."

Companies need to control the types and quantities of materials they purchase, plan which products are to be produced and in what quantities and ensure that they are able to meet current and future customer demand, all at the lowest possible cost. Making a bad decision in any of these areas will make the company lose money. A few examples are given below:

- If a company purchases insufficient quantities of an item used in manufacturing (or the wrong item) it may be unable to meet contract obligations to supply products on time.
- If a company purchases excessive quantities of an item, money is wasted - the excess quantity ties up cash while it remains as stock and may never even be used at all.
- Beginning production of an order at the wrong time can cause customer deadlines to be missed.

MRP is a tool to deal with these problems. It provides answers for several questions:

- *What* items are required?
- *How many* are required?
- *When* are they required?

MRP can be applied both to items that are purchased from outside suppliers and to sub-assemblies, produced internally, that are components of more complex items.

The data that must be considered include:

- The *end item* (or items) being created. This is sometimes called Independent Demand, or Level "0" on BOM (Bill of materials).
- How much is required at a time.
- When the quantities are required to meet demand.
- Shelf life of stored materials.
- Inventory status records. Records of *net materials available* for use already in stock (on hand) and materials on order from suppliers.
- Bills of materials. Details of the materials, components and sub-assemblies required to make each product.

- Planning Data. This includes all the restraints and directions to produce the end items. This includes such items as: Routings, Labor and Machine Standards, Quality and Testing Standards, Pull/Work Cell and Push commands, Lot sizing techniques (i.e. Fixed Lot Size, Lot-For-Lot, Economic Order Quantity), Scrap Percentages, and other inputs.

Outputs

There are two outputs and a variety of messages/reports:

- Output 1 is the "Recommended Production Schedule" which lays out a detailed schedule of the required minimum start and completion dates, with quantities, for each step of the Routing and Bill Of Material required to satisfy the demand from the Master Production Schedule (MPS).
- Output 2 is the "Recommended Purchasing Schedule". This lays out both the dates that the purchased items should be received into the facility AND the dates that the Purchase orders, or Blanket Order Release should occur to match the production schedules.

Messages and Reports:

- Purchase orders. An order to a supplier to provide materials.
- Reschedule notices. These *recommend* cancelling, increasing, delaying or speeding up existing orders.

Problems with MRP systems

The major problem with MRP systems is the integrity of the data. If there are any errors in the inventory data, the [bill of materials] (commonly referred to as 'BOM') data, or the master production schedule, then the outputted data will also be incorrect (colloquially, "GIGO": Garbage In, Garbage Out). Data integrity is also affected by inaccurate cycle count adjustments, mistakes in receiving input and shipping output, scrap not reported, waste, damage, box count errors, supplier container count errors, production reporting errors, and system issues. Many of these types of errors can be minimized by implementing pull systems and using bar code scanning. Most vendors of this type of system recommend at least 99% data integrity for the system to give useful results.

Another major problem with MRP systems is the requirement that the user specify how long it will take a factory to make a product from its component parts (assuming they are all available). Additionally, the system design also assumes that this "lead time" in manufacturing will be the same each time the item is made, without regard to quantity being made, or other items being made simultaneously in the factory.

A manufacturer may have factories in different cities or even countries. It is no good for an MRP system to say that we do not need to order some material because we have plenty thousands of miles away. The overall ERP system needs to be able to organize

inventory and needs by individual factory, and intercommunicate needs in order to enable each factory to redistribute components in order to serve the overall enterprise.

This means that other systems in the enterprise need to work properly both before implementing an MRP system, and into the future. For example systems like variety reduction and engineering which makes sure that product comes out right first time (without defects) must be in place.

Production may be in progress for some part, whose design gets changed, with customer orders in the system for both the old design, and the new one, concurrently. The overall ERP system needs to have a system of coding parts such that the MRP will correctly calculate needs and tracking for both versions. Parts must be booked into and out of stores more regularly than the MRP calculations take place. Note, these other systems can well be manual systems, but must interface to the MRP. For example, a 'walk around' stock intake done just prior to the MRP calculations can be a practical solution for a small inventory (especially if it is an "open store").

The other major drawback of MRP is that takes no account of capacity in its calculations. This means it will give results that are impossible to implement due to manpower or machine or supplier capacity constraints. However this is largely dealt with by MRP II.

Generally, MRP II refers to a system with integrated financials. An MRP II system can include finite / infinite capacity planning. But, to be considered a true MRP II system must also include financials.

In the MRP II (or MRP2) concept, fluctuations in forecast data are taken into account by including simulation of the master production schedule, thus creating a long-term control. A more general feature of MRP2 is its extension to purchasing, to marketing and to finance (integration of all the function of the company), ERP has been the next step.

Solutions to data integrity issues

Bill of material - The best practice is to physically verify the bill of material either at the production site or by un-assembling the product.

Cycle count - The best practice is to determine why a cycle count that increases or decreases inventory has occurred. Find the root cause and correct the problem from occurring again.

Scrap reporting - This can be the most difficult area to maintain with any integrity. Start with isolating the scrap by providing scrap bins at the production site and then record the scrap from the bins on a daily basis. One benefit of reviewing the scrap on site is that preventive action can be taken by the engineering group.

Receiving errors - Manual systems of recording what has been received are error prone. The best practice is to implement the system of receiving by ASN from the supplier. The

supplier sends an ASN (Advanced Shipping Notification). When the components are received into the facility, the ASN is processed and then company labels are created for each line item. The labels are affixed to each container and then scanned into the MRP system. Extra labels reveals a shortage from the shipment and too few labels is an over shipment. Some companies pay by ASN reducing the time in processing accounts payables.

Shipping Errors - The container labels are printed from the shipper. The labels are affixed to the containers in a staging area or when they are loaded on the transport.

Production reporting - The best practice is to use bar code scanning to enter production into inventory. Product that is rejected should be moved to an MRB (material review board) location. Containers that require sorting need to be reversed received.

Replenishment - The best replenishment practice is replacement using bar code scanning, or via pull system. Depending upon the complexity of the product, planners can actually order materials using scanning with a min-max system.

Direct numerical control

Direct numerical control (DNC), also known as **distributed numerical control** (also DNC), is a common manufacturing term for networking CNC machine tools. On some CNC machine controllers, the available memory is too small to contain the machining program (for example machining complex surfaces), so in this case the program is stored in a separate computer and sent *directly* to the machine, one block at a time. If the computer is connected to a number of machines it can *distribute* programs to different machines as required. Usually, the manufacturer of the control provides suitable DNC software. However, if this provision is not possible, some software companies provide DNC applications that fulfill the purpose. DNC networking or DNC communication is always required when CAM programs are to run on some CNC machine control.

1950s-1970s

Programs had to be walked to NC controls, generally on paper tape. NC controls had paper tape readers precisely for this purpose. Many companies were still punching programs on paper tape well into the 1980s, more than twenty-five years after its elimination in the computer industry.

1980s

The focus in the 1980s was mainly on reliably transferring NC programs between a host computer and the control. The Host computers would frequently be Sun Microsystems, HP, Prime, DEC or IBM type computers running a variety of CAD/CAM software. DNC

companies offered machine tool links using rugged proprietary terminals and networks. For example, DLog offered an x86 based terminal, and NCPC had one based on the 6809. The host software would be responsible for tracking and authorising NC program modifications. Depending on program size, for the first time operators had the opportunity to modify programs at the DNC terminal. No time was lost due to broken tapes, and if the software was correctly used, an operator running incorrect or out of date programs became a thing of the past.

Older controls frequently had no port capable of receiving programs such as an RS232 or RS422 connector. In these cases, a device known as a Behind The Reader or BTR card was used. The connection between the control's tape reader and the internal processor was interrupted by a microprocessor based device which emulated the paper tape reader's signals, but which had a serial port connected to the DNC system. As far as the control was concerned, it was receiving from the paper tape unit as it always had; in fact it was the BTR or Reader Emulation card which was transmitting. A switch was frequently added to permit the paper tape reader to be used as a backup.

1990s and beyond

The PC explosion in the late 1980s and early 1990s signalled the end of the road for proprietary DNC terminals. With some exceptions, CNC manufacturers began migrating to PC-based controls running DOS, Windows or OS/2 which could be linked in to existing networks using standard protocols. Customers began migrating away from expensive minicomputer and workstation based CAD/CAM toward more cost-effective PC-based solutions. Users began to demand more from their DNC systems than secure upload/download and editing. PC-based systems which could accomplish these tasks based on standard networks began to be available at minimal or no cost. In some cases, users no longer needed a DNC "expert" to implement shop floor networking, and could do it themselves. However, the task can still be a challenge based on the CNC Control wiring requirements, parameters and NC program format.

To remain competitive, therefore, DNC companies moved their offerings upmarket into DNC Networking, Shop Floor Control or SFC, Manufacturing Execution Systems or MES. These terms encompass concepts such as real-time Machine Monitoring, Graphics, Tool Management, Traveler Management and Scheduling. Instead of merely acting as a repository for programs, DNC systems aim to give operators at the machine an integrated view of all the information (both textual and graphical) they require in order to carry out a manufacturing operation, and give management timely information as to the progress of each step. DNC systems are frequently directly integrated with corporate CAD/CAM, ERP and Computer-aided Process Planning CAPP systems.

Special protocols

A challenge when interfacing into machine tools is that in some cases special protocols are used. Two well-known examples are Mazatrol and Heidenhain. Many DNC systems offer support for these protocols. Another protocol is DNC2 or LSV2 which is found on

Fanuc controls. DNC2 allows advanced interchange of data with the control, such as tooling offsets, tool life information and machine status as well as automated transfer without operator intervention.

Machine monitoring

One of the issues involved in machine monitoring is whether or not it can be accomplished automatically in a practical way. In the 1980s monitoring was typically done by having a menu on the DNC terminal where the operator had to manually indicate what was being done by selecting from a menu, which has obvious drawbacks. There have been advances in passive monitoring systems where the machine condition can be determined by hardware attached in such a way as not to interfere with machine operations (and potentially void warranties). Many modern controls allow external applications to query their status using a special protocol. MTConnect is one prominent attempt to augment the existing world of proprietary systems with some open-source, industry-standard protocols and XML schemas and an ecosystem of massively multiplayer app development and mashups (analogous to that with smartphones) so that these long-sought higher levels of manufacturing business intelligence and workflow automation can be realized.

Alternatives

Smaller facilities will typically use a portable PC, palmtop or laptop to avoid the expense of a fully networked DNC system. In the past Facit Walk Disk and a similar device from Mazak were very popular.

Chapter-6

G-code

G-code is the common name for the most widely used computer numerical control (CNC) programming language, which has many implementations. Used mainly in automation, it is part of computer-aided engineering. This general sense of the term, referring to the language overall (using the mass sense of "code"), is imprecise, because it comes metonymically from the literal sense of the term, referring to one letter address among many in the language (G address, for preparatory commands) and to the specific codes (count sense) that can be formed with it (for example, G00, G01, G28). In fact, every letter of the English alphabet is used somewhere in the language, although some letters' use is less common. Nevertheless, the general sense of the term is indelibly established as the common name of the language. G-code is sometimes called **G programming language**, but most people well versed in CNC work prefer the name *G-code*.

The first implementation of numerical control was developed at the MIT Servomechanisms Laboratory in the early 1950s. In the decades since, many implementations have been developed by many (commercial and noncommercial) organizations. G-code has often been used in these implementations. The main standardized version used in the United States was settled by the Electronic Industries Alliance in the early 1960s. A final revision was approved in February 1980 as *RS274D*. In Europe, the standard *ISO 6983* is often used, although in varied states sometimes used other standards, example *DIN 66025* or PN-73M-55256, PN-93/M-55251 in Poland.

Extensions and variations have been added independently by control manufacturers and machine tool manufacturers, and operators of a specific controller must be aware of differences of each manufacturer's product.

One standardized version of G-code, known as *BCL*, is used only on very few machines.

Some CNC machine manufacturers attempted to overcome compatibility difficulties by standardizing on machine tool controllers built by Fanuc. This semistandardization can be compared to other instances of market dominance, such as with IBM, Intel, or Microsoft. Pros and cons exist, and a wide variety of alternatives are available.

Some CNC machines use "conversational" programming, which is a wizard-like programming mode that either hides G-code or completely bypasses the use of G-code. Some popular examples are Southwestern Industries' ProtoTRAK, Mazak's Mazatrol, Hurco's Ultimax and Mori Seiki's CAPS conversational software.

G-code began as a limited type of language that lacked constructs such as loops, conditional operators, and programmer-declared variables with natural-word-including names (or the expressions in which to use them). It was thus unable to encode logic; it was essentially just a way to "connect the dots" where many of the dots' locations were figured out longhand by the programmer. The latest implementations of G-code include such constructs, creating a language somewhat closer to a high-level programming language. The more a programmer can tell the machine what end result is desired, and leave the intermediate calculations to the machine, the more s/he uses the machine's computational power to full advantage.

Specific codes

G-codes are also called preparatory codes, and are any word in a CNC program that begins with the letter "G". Generally it is a code telling the machine tool what type of action to perform, such as:

- rapid move
- controlled feed move in a straight line or arc
- series of controlled feed moves that would result in a hole being bored, a workpiece cut (routed) to a specific dimension, or a decorative profile shape added to the edge of a workpiece.
- set tool information such as offset.

There are other codes; the type codes can be thought of like registers in a computer.

Letter addresses

Some letter addresses are used only in milling or only in turning; most are used in both. **Bold** below are the letters seen most frequently throughout a program.

Sources: Smid; Green et al.

| Variable | Description | Corollary info |
|-----------------|--|-----------------------|
| A | Absolute or incremental position of A axis (rotational axis around X axis) | |
| B | Absolute or incremental position of B axis (rotational | |

| | | |
|---|---|---|
| | axis around Y axis) | |
| C | Absolute or incremental position of C axis (rotational axis around Z axis) | |
| D | Defines diameter or radial offset used for cutter compensation | |
| E | Precision feedrate for threading on lathes | |
| F | Defines feed rate | |
| G | Address for preparatory commands | G commands often tell the control what kind of motion is wanted (e.g., rapid positioning, linear feed, circular feed, fixed cycle) or what offset value to use. |
| H | Defines tool length offset; Incremental axis corresponding to C axis (e.g., on a turn-mill) | |
| I | Defines arc size in X axis for G02 or G03 arc commands. Also used as a parameter within some fixed cycles. | |
| J | Defines arc size in Y axis for G02 or G03 arc commands. Also used as a parameter within some fixed cycles. | |
| K | Defines arc size in Z axis for G02 or G03 arc commands. Also used as a parameter within some fixed cycles, equal to L address. | |
| L | Fixed cycle loop count; | <i>Fixed cycle loop count:</i> Defines number of repetitions ("loops") of a fixed cycle at <i>each</i> |

| | | |
|----------|--|--|
| | Specification of what register to edit using G10 | position. Assumed to be 1 unless programmed with another integer. Sometimes the K address is used instead of L. With incremental positioning (G91), a series of equally spaced holes can be programmed as a loop rather than as individual positions. <i>G10 use:</i> Specification of what register to edit (work offsets, tool radius offsets, tool length offsets, etc.). |
| M | Miscellaneous function | Action code, auxiliary command; descriptions vary. Many M-codes call for machine functions, which is why people often say that the "M" stands for "machine", although it was not intended to. |
| N | Line (block) number in program; System parameter number to be changed using G10 | <i>Line (block) numbers:</i> Optional, so often omitted. Necessary for certain tasks, such as M99 P address (to tell the control which block of the program to return to if not the default one) or GoTo statements (if the control supports those). N numbering need not increment by 1 (for example, it can increment by 10, 20, or 1000) and can be used on every block or only in certain spots throughout a program. <i>System parameter number:</i> G10 allows changing of system parameters under program control. |
| O | Program name | For example, O4501. |
| P | Serves as parameter address for various G and M codes | <ul style="list-style-type: none"> • With G04, defines dwell time value. • Also serves as a parameter in some canned cycles, representing dwell times or other variables. • Also used in the calling and termination of subprograms. (With M98, it specifies which subprogram to call; with M99, it specifies which block number of the main program to return to.) |
| Q | Peck increment in canned cycles | For example, G73, G83 (peck drilling cycles) |
| R | Defines size of arc radius or defines retract height in canned cycles | |
| S | Defines speed, either spindle speed or surface speed depending on mode | Data type = integer. In G97 mode (which is usually the default), an integer after S is interpreted as a number of rev/min (rpm). In G96 mode (CSS), an integer after S is interpreted as |

| | | |
|---|---|---|
| | | surface speed—sfm (G20) or m/min (G21). On multifunction (turn-mill or mill-turn) machines, which spindle gets the input (main spindle or subspindles) is determined by other M codes. |
| T | Tool selection | To understand how the T address works and how it interacts (or not) with M06, one must study the various methods, such as lathe turret programming, ATC fixed tool selection, ATC random memory tool selection, the concept of "next tool waiting", and empty tools. Programming on any particular machine tool requires knowing which method that machine uses. |
| U | Incremental axis corresponding to X axis (typically only lathe group A controls) Also defines dwell time on some machines (instead of "P" or "X"). | In these controls, X and U obviate G90 and G91, respectively. On these lathes, G90 is instead a fixed cycle address for roughing. |
| V | Incremental axis corresponding to Y axis | Until the 2000s, the V address was very rarely used, because most lathes that used U and W didn't have a Y-axis, so they didn't use V. (Green et al 1996 did not even list V in their table of addresses.) That is still often the case, although the proliferation of live lathe tooling and turn-mill machining has made V address usage less rare than it used to be (Smid 2008 shows an example). |
| W | Incremental axis corresponding to Z axis (typically only lathe group A controls) | In these controls, Z and W obviate G90 and G91, respectively. On these lathes, G90 is instead a fixed cycle address for roughing. |
| X | Absolute or incremental position of X axis. Also defines dwell time on some machines (instead of "P" or "U"). | |
| Y | Absolute or incremental position | |

| | | |
|----------|--|--|
| | of Y axis | |
| Z | Absolute or incremental position of Z axis | The main spindle's axis of rotation often determines which axis of a machine tool is labeled as Z. |

List of G-codes commonly found on Fanuc and similarly designed controls

| Code | Description | Milling (M) | Turning (T) | Corollary info |
|------|-------------------------|---------------|---------------|---|
| G00 | Rapid positioning | M | T | On 2- or 3-axis moves, G00 (unlike G01) traditionally does not necessarily move in a single straight line between start point and end point. It moves each axis at its max speed until its vector is achieved. Shorter vector usually finishes first (given similar axis speeds). This matters because it may yield a dog-leg or hockey-stick motion, which the programmer needs to consider depending on what obstacles are nearby, to avoid a crash. Some machines offer interpolated rapids as a feature for ease of programming (safe to assume a straight line). |
| G01 | Linear interpolation | M | T | The most common workhorse code for feeding during a cut. The program specs the start and end points, and the control automatically calculates (interpolates) the intermediate points to pass through that will yield a straight line (hence "linear"). The control then calculates the angular velocities at which to turn the axis leadscrews. The computer performs thousands of calculations per second. Actual machining takes place with given feed on linear path. |
| G02 | Circular interpolation, | M | T | Cannot start G41 or G42 in G02 or G03 modes. Must already be |

| | | | | |
|---------------|---|---|---|--|
| | clockwise | | | compensated in earlier G01 block. |
| G03 | Circular interpolation, counterclockwise | M | T | Cannot start G41 or G42 in G02 or G03 modes. Must already be compensated in earlier G01 block. |
| G04 | Dwell | M | T | Takes an address for dwell period (may be X, U, or P) |
| G05 P10000 | High-precision contour control (HPCC) | M | | Uses a deep look-ahead buffer and simulation processing to provide better axis movement acceleration and deceleration during contour milling |
| G05.1 Q1. | Ai Nano contour control | M | | Uses a deep look-ahead buffer and simulation processing to provide better axis movement acceleration and deceleration during contour milling |
| G07 | Imaginary axis designation | M | | |
| G09 | Exact stop check | M | T | |
| G10 | Programmable data input | M | T | |
| G11 | Data write cancel | M | T | |
| G12 | Full-circle interpolation, clockwise | M | | Fixed cycle for ease of programming 360° circular interpolation with blend-radius lead-in and lead-out. Not standard on Fanuc controls. |
| G13 | Full-circle interpolation, counterclockwise | M | | Fixed cycle for ease of programming 360° circular interpolation with blend-radius lead-in and lead-out. Not standard on Fanuc controls. |
| G17 | XY plane selection | M | | |
| G18 | ZX plane selection | M | T | On most CNC lathes (built 1960s to 2000s), ZX is the only available plane, so no G17 to G19 codes are used. This is now changing as the era begins in which live tooling, multitask/multifunction, and mill-turn/turn-mill gradually become the "new normal". But the simpler, traditional form factor |

| | | | | |
|-----|---|---|---|---|
| | | | | will probably not disappear—just move over to make room for the newer configurations. |
| G19 | YZ plane selection | M | | |
| G20 | Programming in inches | M | T | Somewhat uncommon except in USA and (to lesser extent) Canada and UK. However, in the global marketplace, competence with both G20 and G21 always stands some chance of being necessary at any time. The usual minimum increment in G20 is one ten-thousandth of an inch (0.0001"), which is a larger distance than the usual minimum increment in G21 (one thousandth of a millimeter, .001 mm, that is, one micrometre). This physical difference sometimes favors G21 programming. |
| G21 | Programming in millimeters (mm) | M | T | Prevalent worldwide. However, in the global marketplace, competence with both G20 and G21 always stands some chance of being necessary at any time. |
| G28 | Return to home position (machine zero, aka machine reference point) | M | T | Takes X Y Z addresses which define the intermediate point that the tool tip will pass through on its way home to machine zero. They are in terms of part zero (aka program zero), NOT machine zero. |
| G30 | Return to secondary home position (machine zero, aka machine reference point) | M | T | Takes a P address specifying <i>which</i> machine zero point is desired, <i>if</i> the machine has several secondary points (P1 to P4). Takes X Y Z addresses which define the intermediate point that the tool tip will pass through on its way home to machine zero. They are in terms of part zero (aka program zero), NOT machine zero. |
| G31 | Skip function (used | M | | |

| | | | | |
|-----|--|---|---|--|
| | for probes and tool length measurement systems) | | | |
| G32 | Single-point threading, longhand style (if not using a cycle, e.g., G76) | | T | Similar to G01 linear interpolation, except with automatic spindle synchronization for single-point threading. |
| G33 | Constant-pitch threading | M | | |
| G33 | Single-point threading, longhand style (if not using a cycle, e.g., G76) | | T | Some lathe controls assign this mode to G33 rather than G32. |
| G34 | Variable-pitch threading | M | | |
| G40 | Tool radius compensation off | M | T | Cancels G41 or G42. |
| G41 | Tool radius compensation left | M | T | Milling: Given righthand-helix cutter and M03 spindle direction, G41 corresponds to climb milling (down milling). Takes an address (D or H) that calls an offset register value for radius. Turning: Often needs no D or H address on lathes, because whatever tool is active automatically calls its geometry offsets with it. (Each turret station is bound to its geometry offset register.) |
| G42 | Tool radius compensation right | M | T | Similar corollary info as for G41. Given righthand-helix cutter and M03 spindle direction, G42 corresponds to conventional milling (up milling). |
| G43 | Tool height offset compensation negative | M | | Takes an address, usually H, to call the tool length offset register value. The value is <i>negative</i> because it will be <i>added</i> to the gauge line position. G43 is the |

| | | | | |
|-----|--|---|---|---|
| | | | | commonly used version (vs G44). |
| G44 | Tool height offset compensation positive | M | | Takes an address, usually H, to call the tool length offset register value. The value is <i>positive</i> because it will be <i>subtracted</i> from the gauge line position. G44 is the seldom-used version (vs G43). |
| G45 | Axis offset single increase | M | | |
| G46 | Axis offset single decrease | M | | |
| G47 | Axis offset double increase | M | | |
| G48 | Axis offset double decrease | M | | |
| G49 | Tool length offset compensation cancel | M | | Cancels G43 or G44. |
| G50 | Define the maximum spindle speed | | T | Takes an S address integer which is interpreted as rpm. Without this feature, G96 mode (CSS) would rev the spindle to "wide open throttle" when closely approaching the axis of rotation. |
| G50 | Scaling function cancel | M | | |
| G50 | Position register (programming of vector from part zero to tool tip) | | T | Position register is one of the original methods to relate the part (program) coordinate system to the tool position, which indirectly relates it to the machine coordinate system, the only position the control really "knows". Not commonly programmed anymore because G54 to G59 (WCSs) are a better, newer method. Called via G50 for turning, G92 for milling. Those G addresses also have alternate meanings (which see). Position register can still be useful for datum shift programming. |
| G52 | Local coordinate | M | | Temporarily shifts program zero |

| | | | | |
|-----------------|---|---|---|--|
| | system (LCS) | | | to a new location. This simplifies programming in some cases. |
| G53 | Machine coordinate system | M | T | Takes absolute coordinates (X,Y,Z,A,B,C) with reference to machine zero rather than program zero. Can be helpful for tool changes. Nonmodal and absolute only. Subsequent blocks are interpreted as "back to G54" even if it is not explicitly programmed. |
| G54 to G59 | Work coordinate systems (WCSs) | M | T | Have largely replaced position register (G50 and G92). Each tuple of axis offsets relates program zero directly to machine zero. Standard is 6 tuples (G54 to G59), with optional extensibility to 48 more via G54.1 P1 to P48. |
| G54.1 P1 to P48 | Extended work coordinate systems | M | T | Up to 48 more WCSs besides the 6 provided as standard by G54 to G59. Note floating-point extension of G-code data type (formerly all integers). Other examples have also evolved (e.g., G84.2). Modern controls have the hardware to handle it. |
| G70 | Fixed cycle, multiple repetitive cycle, for finishing (including contours) | | T | |
| G71 | Fixed cycle, multiple repetitive cycle, for roughing (Z-axis emphasis) | | T | |
| G72 | Fixed cycle, multiple repetitive cycle, for roughing (X-axis emphasis) | | T | |
| G73 | Fixed cycle, multiple repetitive cycle, for roughing, with pattern repetition | | T | |

| | | | | |
|-----|--|---|---|--|
| G73 | Peck drilling cycle for milling - high-speed (NO full retraction from pecks) | M | | Retracts only as far as a clearance increment (system parameter). For when chipbreaking is the main concern, but chip clogging of flutes is not. |
| G74 | Peck drilling cycle for turning | | T | |
| G74 | Tapping cycle for milling, lefthand thread, M04 spindle direction | M | | |
| G75 | Peck grooving cycle for turning | | T | |
| G76 | Fine boring cycle for milling | M | | |
| G76 | Threading cycle for turning, multiple repetitive cycle | | T | |
| G80 | Cancel canned cycle | M | T | Milling: Cancels all cycles such as G73, G83, G88, etc. Z-axis returns either to Z-initial level or R-level, as programmed (G98 or G99, respectively). Turning: Usually not needed on lathes, because a new group-1 G address (G00 to G03) cancels whatever cycle was active. |
| G81 | Simple drilling cycle | M | | No dwell built in |
| G82 | Drilling cycle with dwell | M | | Dwells at hole bottom (Z-depth) for the number of milliseconds specified by the P address. Good for when hole bottom finish matters. |
| G83 | Peck drilling cycle (full retraction from pecks) | M | | Returns to R-level after each peck. Good for clearing flutes of chips. |
| G84 | Tapping cycle, righthand thread, M03 spindle direction | M | | |

| | | | | |
|-------|--|---|-------|---|
| G84.2 | Tapping cycle, righthand thread, M03 spindle direction, rigid toolholder | M | | |
| G90 | Absolute programming | M | T (B) | Positioning defined with reference to part zero. Milling: Always as above. Turning: Sometimes as above (Fanuc group type B and similarly designed), but on most lathes (Fanuc group type A and similarly designed), G90/G91 are not used for absolute/incremental modes. Instead, U and W are the incremental addresses and X and Z are the absolute addresses. On these lathes, G90 is instead a fixed cycle address for roughing. |
| G90 | Fixed cycle, simple cycle, for roughing (Z-axis emphasis) | | T (A) | When not serving for absolute programming (above) |
| G91 | Incremental programming | M | T (B) | Positioning defined with reference to previous position. Milling: Always as above. Turning: Sometimes as above (Fanuc group type B and similarly designed), but on most lathes (Fanuc group type A and similarly designed), G90/G91 are not used for absolute/incremental modes. Instead, U and W are the incremental addresses and X and Z are the absolute addresses. On these lathes, G90 is a fixed cycle address for roughing. |
| G92 | Position register (programming of vector from part zero to tool tip) | M | T (B) | Same corollary info as at G50 position register. Milling: Always as above. Turning: Sometimes as above (Fanuc group type B and similarly designed), but on most lathes (Fanuc group type A and similarly designed), position register is G50. |

| | | | | |
|-----|---|---|-------|--|
| G92 | Threading cycle, simple cycle | | T (A) | |
| G94 | Feedrate per minute | M | T (B) | On group type A lathes, feedrate per minute is G98. |
| G94 | Fixed cycle, simple cycle, for roughing (X-axis emphasis) | | T (A) | When not serving for feedrate per minute (above) |
| G95 | Feedrate per revolution | M | T (B) | On group type A lathes, feedrate per revolution is G99. |
| G96 | Constant surface speed (CSS) | | T | Varies spindle speed automatically to achieve a constant surface speed. Takes an S address integer, which is interpreted as sfm in G20 mode or as m/min in G21 mode. |
| G97 | Constant spindle speed | M | T | Takes an S address integer, which is interpreted as rev/min (rpm). The default speed mode per system parameter if no mode is programmed. |
| G98 | Return to initial Z level in canned cycle | M | | |
| G98 | Feedrate per minute (group type A) | | T (A) | Feedrate per minute is G94 on group type B. |
| G99 | Return to R level in canned cycle | M | | |
| G99 | Feedrate per revolution (group type A) | | T (A) | Feedrate per revolution is G95 on group type B. |

List of M-codes commonly found on Fanuc and similarly designed controls

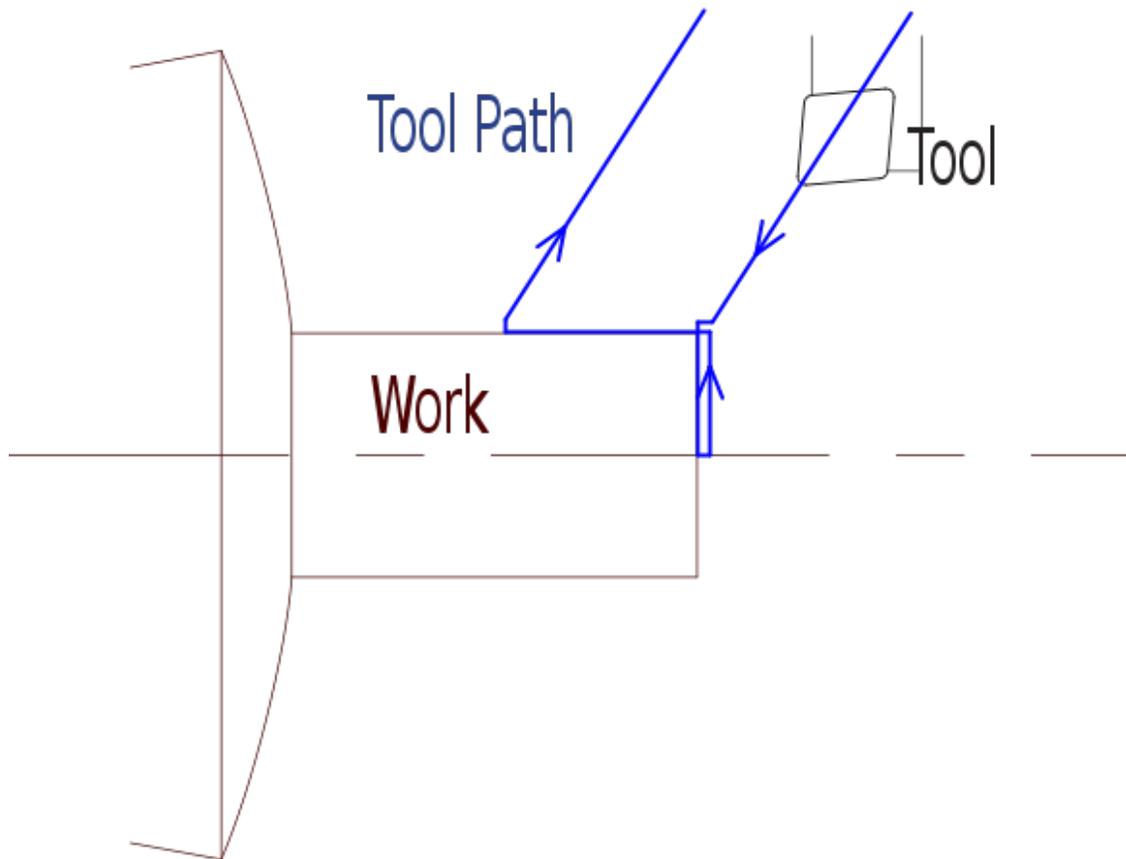
| Code | Description | Milling (M) | Turning (T) | Corollary info |
|------|-----------------|-------------|-------------|---|
| M00 | Compulsory stop | M | T | Non-optional—machine will always stop upon reaching M00 in the program execution. |
| M01 | Optional stop | M | T | Machine will only stop at M01 if operator has pushed the optional stop button. |

| | | | | |
|-----|--|---|---------------|---|
| M02 | End of program | M | T | No return to program top; may or may not reset register values. |
| M03 | Spindle on (clockwise rotation) | M | T | How to objectively define "clockwise" or "counterclockwise" (CW or CCW) when the same rotating cylinder can be viewed as either CW or CCW, depending on which end of it the viewer is looking from? The answer links back to the right-hand rule. Right-hand-helix screws moving in the tightening direction (and right-hand-helix flutes spinning in the cutting direction) are defined as moving in the M03 direction, and are labeled "clockwise" by convention. The M03 direction is always M03 regardless of local vantage point and local CW/CCW distinction. |
| M04 | Spindle on (counterclockwise rotation) | M | T | |
| M05 | Spindle stop | M | T | |
| M06 | Automatic tool change (ATC) | M | T (sometimes) | Many lathes do not use M06 because the T address itself indexes the turret. To understand how the T address works and how it interacts (or not) with M06, one must study the various methods, such as lathe turret programming, ATC fixed tool selection, ATC random memory tool selection, the concept of "next tool waiting", and empty tools. Programming on any particular machine tool requires knowing which method that machine uses. |
| M07 | Coolant on (mist) | M | T | |
| M08 | Coolant on (flood) | M | T | |
| M09 | Coolant off | M | T | |

| | | | | |
|-----|--|---|---|---|
| M10 | Pallet clamp on | M | | For machining centers with pallet changers |
| M11 | Pallet clamp off | M | | For machining centers with pallet changers |
| M13 | Spindle on (clockwise rotation) and coolant on (flood) | M | | This one M-code does the work of both M03 and M08. It is not unusual for specific machine models to have such combined commands, which make for shorter, more quickly written programs. |
| M19 | Spindle orientation | M | T | Spindle orientation is more often called within cycles (automatically) or during setup (manually), but it is also available under program control via M19 . The abbreviation OSS (oriented spindle stop) may be seen in reference to an oriented stop within cycles. |
| M21 | Mirror, X-axis | M | | |
| M21 | Tailstock forward | | T | |
| M22 | Mirror, Y-axis | M | | |
| M22 | Tailstock backward | | T | |
| M23 | Mirror OFF | M | | |
| M23 | Thread gradual pullout ON | | T | |
| M24 | Thread gradual pullout OFF | | T | |
| M30 | End of program with return to program top | M | T | |
| M41 | Gear select - gear 1 | | T | |
| M42 | Gear select - gear 2 | | T | |
| M43 | Gear select - gear 3 | | T | |
| M44 | Gear select - gear 4 | | T | |
| M48 | Feedrate override allowed | M | T | |

| | | | | |
|-----|----------------------------------|---|---|--|
| M49 | Feedrate override NOT allowed | M | T | This rule is also called (automatically) within tapping cycles or single-point threading cycles, where feed is precisely correlated to speed. Same with spindle speed override and feed hold button. |
| M60 | Automatic pallet change (APC) | M | | For machining centers with pallet changers |
| M98 | Subprogram call | M | T | Takes an address P to specify which subprogram to call, for example, "M98 P8979" calls subprogram O8979. |
| M99 | Subprogram end | M | T | Usually placed at end of subprogram, where it returns execution control to the main program. The default is that control returns to the block following the M98 call in the main program. Return to a different block number can be specified by a P address. M99 can also be used in main program with block skip for endless loop of main program on bar work on lathes (until operator toggles block skip). |

Example program



Tool Path for program

This is a generic program that demonstrates the use of G-Code to turn a 1" diameter X 1" long part. Assume that a bar of material is in the machine and that the bar is slightly oversized in length and diameter and that the bar protrudes by more than 1" from the face of the chuck. (Caution: This is generic, it might not work on any real machine! Pay particular attention to point 5 below.)

Sample

| Line | Code | Description |
|-------|----------------------------|--|
| O4968 | | (Sample face and turn program) |
| N01 | M216 | (Turn on load monitor) |
| N02 | G20 G90 G54 D200 G40 | (Inch units. Absolute mode. Call work offset values. Moving coordinate system to the location specified in the register D200. Cancel any existing tool radius offset.) |
| N03 | G50 S2000 | (Set maximum spindle speed rev/min - preparing for G96 CSS coming soon) |
| N04 | M01 | (Optional stop) |

N05 T0300 (Index turret to tool 3. Clear wear offset (00).)
G96 S854 (Constant surface speed [automatically varies the spindle
N06 M42 M03 speed], 854 sfm, select spindle gear, start spindle CW rotation,
M08 turn on the coolant flood)
(Call tool radius offset. Call tool wear offset. Rapid feed to a
N07 G41 G00 point *about* 0.100" from the end of the bar [not counting
X1.1 Z1.1 0.005" or 0.006" that the bar-pull-and-stop sequence is set up
T0303 to leave as a stock allowance for facing off] and 0.050" from
the side)
N08 G01 Z1.0 (Feed in horizontally until the tool is standing 1" from the
F.05 datum i.e. program Z-zero)
N09 X-0.002 (Feed down until the tool is slightly past center, thus facing the
end of the bar)
N10 G00 Z1.1 (Rapid feed 0.1" away from the end of the bar - clear the part)
N11 X1.0 (Rapid feed up until the tool is standing at the finished OD)
N12 G01 Z0.0 (Feed in horizontally cutting the bar to 1" diameter all the way
F.05 to the datum, feeding at 0.050" per revolution)
N13 G00 X1.1 (Clear the part, stop the spindle, turn off the coolant)
M05 M09
N14 G91 G28 (Home X axis - return to machine X-zero passing through no
X0 intermediate X point [incremental X0])
N15 G91 G28 (Home Z axis - return to machine Z-zero passing through no
Z0 intermediate Z point [incremental Z0])
N16 G90 (Return to absolute mode. Turn off load monitor)
M215
N17 M30 (Program stop, rewind to beginning of program)
%

Several points to note:

1. There is room for some programming style, even in this short program. The grouping of codes in line N06 could have been put on multiple lines. Doing so may have made it easier to follow program execution.
2. Many codes are "modal", meaning that they stay in effect until they are cancelled or replaced by a contradictory code. For example, once variable speed cutting (CSS) had been selected (G96), it stayed in effect until the end of the program. In operation, the spindle speed would increase as the tool neared the center of the work in order to maintain a constant surface speed. Similarly, once rapid feed was selected (G00), all tool movements would be rapid until a feed rate code (G01, G02, G03) was selected.
3. It is common practice to use a load monitor with CNC machinery. The load monitor will stop the machine if the spindle or feed loads exceed a preset value that is set during the set-up operation. The job of the load monitor is to prevent

- machine damage in the event of tool breakage or a programming mistake. On small or hobby machines, it can warn of a tool that is becoming dull and needs to be replaced or sharpened.
4. It is common practice to bring the tool in rapidly to a "safe" point that is close to the part - in this case 0.1" away - and then start feeding the tool. How close that "safe" distance is, depends on the skill of the programmer and maximum material condition for the raw stock.
 5. If the program is wrong, there is a high probability that the machine will *crash*, or ram the tool into the part under high power. This can be costly, especially in newer machining centers. It is possible to intersperse the program with optional stops (M01 code) which allow the program to be run piecemeal for testing purposes. The optional stops remain in the program but they are skipped during the normal running of the machine. Thankfully, most CAD/CAM software ships with CNC simulators that will display the movement of the tool as the program executes. Many modern CNC machines also allow programmers to execute the program in a simulation mode and observe the operating parameters of the machine at a particular execution point. This enables programmers to discover semantic errors (as opposed to syntax errors) before losing material or tools to an incorrect program. Depending on the size of the part, wax blocks may be used for testing purposes as well.
 6. For pedagogical purposes, line numbers have been included in the program above. They are usually not necessary for operation of a machine, so they are seldom used in industry. However, if branching or looping statements are used in the code, then line numbers may well be included as the target of those statements (e.g. GOTO N99).
 7. Some machines do not allow multiple M codes in the same line.

Programming environments

G-code's programming environments have evolved in parallel with those of general programming—from the earliest environments (e.g., writing a program with a pencil, typing it into a tape puncher) to the latest environments that stack computer-aided design (CAD), computer-aided manufacturing (CAM), and richly featured G-code editors. (G-code editors are analogous to XML editors, using colors and indents semantically [plus other features] to aid the user in ways that basic text editors can't. CAM packages are analogous to IDEs in general programming.)

Two high-level paradigm shifts have been (1) abandoning "manual programming" (with nothing but a pencil or text editor and a human mind) for CAM software systems that generate G-code automatically via postprocessors (analogous to the development of visual techniques in general programming), and (2) abandoning hardcoded constructs for parametric ones (analogous to the difference in general programming between hardcoding a constant into an equation versus declaring it a variable and assigning new values to it at will). Macro (parametric) CNC programming uses human-friendly variable names, relational operators, and loop structures much as general programming does, to capture information and logic with machine-readable semantics. Whereas older manual

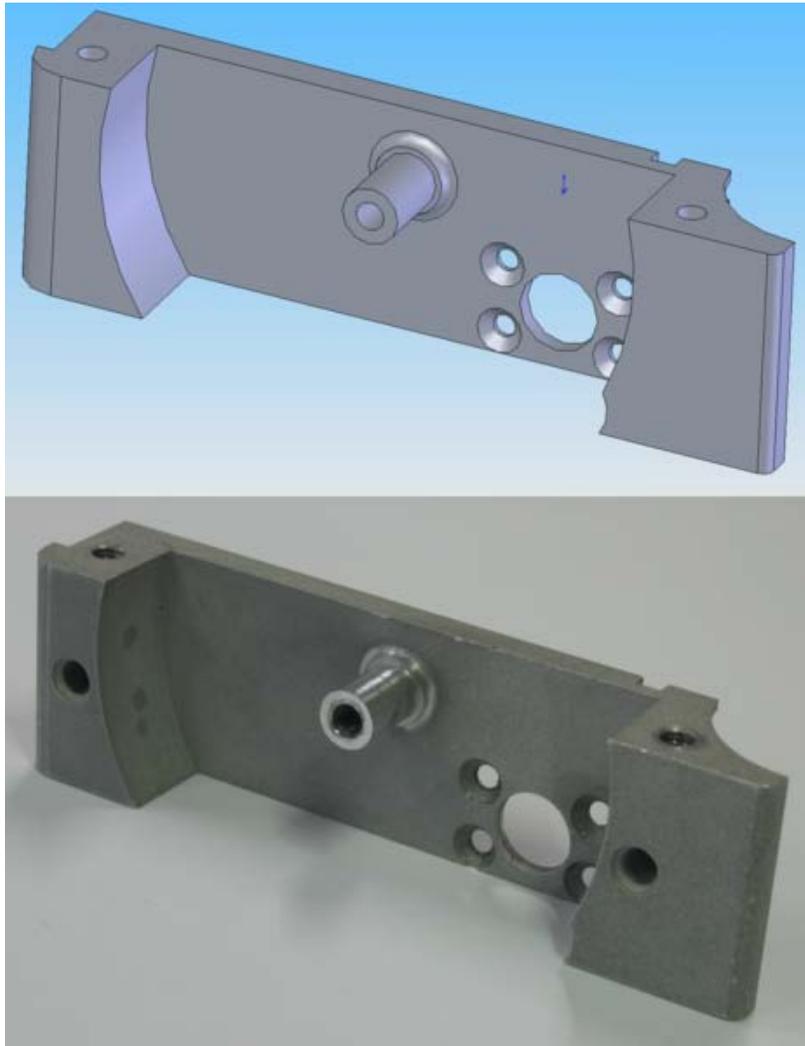
CNC programming could only describe particular instances of parts in numeric form, parametric CAM programming describes abstractions which can be flowed with ease into a wide variety of instances. The difference is analogous to creating text as bitmaps versus using character encoding and glyphs, or to the way that HTML passed through a phase of using content markup for presentation purposes, then matured toward the CSS model. In all of these cases, a higher layer of abstraction was introduced in order to pursue what was missing semantically.

STEP-NC reflects the same theme, which can be viewed as yet another step along a path that started with the development of machine tools, jigs and fixtures, and numerical control, which all sought to "build the skill into the tool". Recent developments of G-code and STEP-NC aim to build the information and semantics into the tool. The idea itself is not new; from the beginning of numerical control, the concept of an end-to-end CAD/CAM environment was the goal of such early technologies as DAC-1 and APT. Those efforts were fine for huge corporations like GM and Boeing. However, for small and medium enterprises, there had to be an era in which the simpler implementations of NC, with relatively primitive "connect-the-dots" G-code and manual programming, ruled the day until CAD/CAM could improve and disseminate throughout the economy.

MTConnect aims to connect machine tools to each other and to other systems in the factory with a much higher level of interaction and capability than has previously existed. Although direct numerical control (DNC) has been networking CNC machine tools to the rest of the enterprise for years, the ability of the various kinds of machines "to talk to each other" has been rather limited in practice (more often than not), compared to the theoretical possibilities. DNC has a lot more potential than just "sending a program to a machine tool over a wire instead of on a tape or disk." But unlocking that potential has been a slow process so far. By creating open-source industry standards (e.g., APIs, XML schemas), MTConnect hopes to spur greater interaction between proprietary systems and a wider developer community. MT Connect might be for manufacturing-segment IT what the Web and app stores have been for other IT domains (commerce, personal, telecoms): a way to bridge the gap between the traditional corporate development environment and the hacker universe. Just as hackers may bring novel uses to smartphones, tablet computers, or Kinects, perhaps they will soon be able to innovate similarly in manufacturing. The enthusiasm of the additive manufacturing community shows how much interest hackers and inventors have in such endeavors.

Chapter-7

Computer-aided Manufacturing



CAD model and CNC machined part

Computer-aided manufacturing (CAM) is the use of computer software to control machine tools and related machinery in the manufacturing of workpieces. This is not the only definition for CAM, but it is the most common; CAM may also refer to the use of a computer to assist in all operations of a manufacturing plant, including planning, management, transportation and storage. Its primary purpose is to create a faster production process and components and tooling with more precise dimensions and material consistency, which in some cases, uses only the required amount of raw material (thus minimizing waste), while simultaneously reducing energy consumption.

CAM is a subsequent computer-aided process after computer-aided design (CAD) and sometimes computer-aided engineering (CAE), as the model generated in CAD and verified in CAE can be input into CAM software, which then controls the machine tool.

Overview



Chrome-cobalt disc with dental implants manufactured using CAM

Traditionally, CAM has been considered as a numerical control (NC) programming tool, wherein two-dimensional (2-D) or three-dimensional (3-D) models of components generated in CAD software are used to generate G-code to drive computer numerically controlled (CNC) machine tools. Simple designs such as bolt circles or basic contours do not necessitate importing a CAD file.

As with other “Computer-Aided” technologies, CAM does not eliminate the need for skilled professionals such as manufacturing engineers, NC programmers, or machinists. CAM, in fact, leverages both the value of the most skilled manufacturing professionals through advanced productivity tools, while building the skills of new professionals through visualization, simulation and optimization tools.

History

The first commercial applications of CAM were in large companies in the automotive and aerospace industries for example UNISURF in 1971 at Renault for car body design and tooling.

Historically, CAM software was seen to have several shortcomings that necessitated an overly high level of involvement by skilled CNC machinists. Fallows created the first CAM software but this had severe shortcomings and was promptly taken back into the developing stage. CAM software would output code for the least capable machine, as each machine tool control added on to the standard G-code set for increased flexibility. In some cases, such as improperly set up CAM software or specific tools, the CNC machine required manual editing before the program will run properly. None of these issues were so insurmountable that a thoughtful engineer or skilled machine operator could not overcome for prototyping or small production runs; G-Code is a simple language. In high production or high precision shops, a different set of problems were encountered where an experienced CNC machinist must both hand-code programs and run CAM software.

Integration of CAD with other components of CAD/CAM/CAE Product lifecycle management (PLM) environment requires an effective CAD data exchange. Usually it had been necessary to force the CAD operator to export the data in one of the common data formats, such as IGES or STL, that are supported by a wide variety of software. The output from the CAM software is usually a simple text file of G-code, sometimes many thousands of commands long, that is then transferred to a machine tool using a direct numerical control (DNC) program.

CAM packages could not, and still cannot, reason as a machinist can. They could not optimize toolpaths to the extent required of mass production. Users would select the type of tool, machining process and paths to be used. While an engineer may have a working knowledge of g-code programming, small optimization and wear issues compound over time. Mass-produced items that require machining are often initially created through casting or some other non-machine method. This enables hand-written, short, and highly optimized g-code that could not be produced in a CAM package.

At least in the United States, there is a shortage of young, skilled machinists entering the workforce able to perform at the extremes of manufacturing; high precision and mass production. As CAM software and machines become more complicated, the skills required of a machinist or machine operator advance to approach that of a computer programmer and engineer rather than eliminating the CNC machinist from the workforce.

Typical areas of concern:

- High Speed Machining, including streamlining of tool paths
- Multi-function Machining
- 5 Axis Machining
- Feature recognition and machining
- Automation of Machining processes
- Ease of Use

Overcoming historical shortcomings

Over time, the historical shortcomings of CAM are being attenuated, both by providers of niche solutions and by providers of high-end solutions. This is occurring primarily in three arenas:

1. Ease of use
2. Manufacturing complexity
3. Integration with PLM and the extended enterprise

Ease in use

For the user who is just getting started as a CAM user, out-of-the-box capabilities providing Process Wizards, templates, libraries, machine tool kits, automated feature based machining and job function specific tailorable user interfaces build user confidence and speed the learning curve.

User confidence is further built on 3D visualization through a closer integration with the 3D CAD environment, including error-avoiding simulations and optimizations.

Manufacturing complexity

The manufacturing environment is increasingly complex. The need for CAM and PLM tools by the manufacturing engineer, NC programmer or machinist is similar to the need for computer assistance by the pilot of modern aircraft systems. The modern machinery cannot be properly used without this assistance.

Today's CAM systems support the full range of machine tools including: turning, 5 axis machining and wire EDM. Today's CAM user can easily generate streamlined tool paths, optimized tool axis tilt for higher feed rates and optimized Z axis depth cuts as well as driving non-cutting operations such as the specification of probing motions.

Integration with PLM and the extended enterprise
LM to integrate manufacturing with enterprise operations from concept through field support of the finished product.

To ensure ease of use appropriate to user objectives, modern CAM solutions are scalable from a stand-alone CAM system to a fully integrated multi-CAD 3D solution-set. These solutions are created to meet the full needs of manufacturing personnel including part planning, shop documentation, resource management and data management and exchange.

Machining process

Most machining progresses through four stages, each of which is implemented by a variety of basic and sophisticated strategies, depending on the material and the software available. The stages are:

Roughing

This process begins with raw stock, known as billet, and cuts it very roughly to shape of the final model. In milling, the result often gives the appearance of terraces, because the strategy has taken advantage of the ability to cut the model horizontally. Common strategies are zig-zag clearing, offset clearing, plunge roughing, rest-roughing.

Semi-finishing

This process begins with a roughed part that unevenly approximates the model and cuts to within a fixed offset distance from the model. The semi-finishing pass must leave a small amount of material so the tool can cut accurately while finishing, but not so little that the tool and material deflect instead of shearing. Common strategies are raster passes, waterline passes, constant step-over passes, pencil milling.

Finishing

Finishing involves a slow pass across the material in very fine steps to produce the finished part. In finishing, the step between one pass and another is minimal. Feed rates are low and spindle speeds are raised to produce an accurate surface.

Contour milling

In milling applications on hardware with five or more axes, a separate finishing process called contouring can be performed. Instead of stepping down in fine-grained increments to approximate a surface, the workpiece is rotated to make the cutting surfaces of the tool tangent to the ideal part features. This produces an excellent surface finish with high dimensional accuracy.

Software

The 20 largest CAM software products and companies, by vendor revenues in year 2009 are, sorted alphabetically:

- BOBCAD-CAM from BobCAD-CAM
- CATIA from Dassault Systèmes
- CAM-Tool from C & G Systems
- Cimatron from Cimatron group
- Dynavista from Nihon Unisys
- Edgecam from Planit
- Esprit from DP Technoogy
- HyperMill from Open Mind
- Mastercam from CNC Software
- NX from Siemens PLM Software
- Powermill from Delcam

- Pro/E from PTC
- SolidCAM from SolidCAM
- Space E from NTTD
- SurfCAM from Surfware
- TopCAM from Missler
- Tebis from Tebis AG
- VisiCAM from Vero
- VisualMILL from MecSoft
- Vericut from CGtech
- WorkNC from SESCOI

Chapter-8

CAD/CAM Dentistry and Integrated Computer-Aided Manufacturing

CAD/CAM Dentistry



Chrome-cobalt disc with bridges and crowns manufactured using WorkNC Dental CAD/CAM

CAD/CAM Dentistry, (Computer-Aided Design and Computer-Aided Manufacturing in Dentistry), is an area of dentistry utilizing CAD/CAM technologies to produce different types of dental restorations, including crowns, crownlays, veneers, inlays and onlays, fixed bridges, dental implant restorations and orthodontic appliances.

History

CAD/CAM usage in Dentistry began in the mid 1980's. Initially, this computerized technology was cumbersome and basically a novelty, requiring inordinate amounts of time to produce a viable product. This reality limited its practical usefulness to the dental laboratory setting, rather than the dental practice as time constraints precluded chairside (in the dental office) usage. As adjunctive techniques, software, and available materials improved over time, CAD/CAM usage in the dental practice setting became a reality. Thus "chairside" CAD/CAM technology affords the practitioner the ability to produce aesthetic, well-fitting prosthetic dental restorations on-site in a matter of hours.

The concept of chairside CAD/CAM restoration differs from conventional dentistry in that the prosthesis is typically luted or bonded in place the same day, whereas conventional dental prosthesis of larger size such as crowns have temporaries placed for several weeks while a dental laboratory may produce the restoration off-site. The patient returns weeks later, the temporary is removed, and then the laboratory made crown is then cemented or bonded in place. Because the CAD/CAM restoration is bonded the same day, the principles applied in preparing the tooth for a crown or other prosthesis are typically more conservative since a temporary is just that, temporary, and the cements used to hold them in place must be weak enough to allow removal. As such, the teeth need to be prepared such that the geometry allows for adequate retention of the temporary in the conventional sense. Since retention can only be achieved with increased surface area, which translates into the removal of extra tooth structure, the conventional restoration typically requires the removal of more tooth structure. With the CAD/CAM restoration, lack of temporization makes physical retention unnecessary since the permanent bonding of the restoration occurs within hours, usually while the patient waits. Consequently, less unnecessary tooth removal is usually the norm.

Process

Typically CAD/CAM restorations are milled from solid blocks of white composite resin or porcelain matching the shade of the restored tooth. After the problem is removed from the diseased tooth, an image is taken of the resulting defect. This image draws the data into a computer and proprietary software is used to create a seal for the defect within the program, essentially creating a virtual restoration. The software then sends this virtual data to a milling chamber by the dentist where the seal for the defect (the dental restoration) is carved out of a solid block of composite resin or porcelain. The resultant restoration can then be adjusted in the patients mouth and bonded in place. If porcelain is used, practitioners usually will treat the restoration with stains and glazes and subsequent heat treatments to both beautify and strengthen the definitive restoration prior to bonding. Following acid etching of both the underside of the restoration and the topside of the

tooth itself (this microscopically increases surface area on both opposing surfaces), composite resin materials are then used to fuse the resultant restoration to the tooth, completing the restoration process.

As milling accuracies are typically accurate to 50 micrometres, CAD/CAM dental restorations are comparable in fit to traditionally fabricated laboratory made dental restorations. Their uniform, monolithic nature blends extremely well with natural tooth structure in contrast to traditional dental restorations where porcelain is fused onto a metal substructure. As these dark metal substructures are not conducive to a natural appearance, milled CAD/CAM restorations are typically more esthetically pleasing to the patient. Since these restorations contain no metal to block subsequent x-rays, dentists are able to keep track of potential decay underneath a CAD/CAM restoration whereas conventional porcelain to metal or traditional gold crowns block x-ray radiation, disallowing such an evaluation over time.

As CAD/CAM enables dental practitioners to effect a final seal in as little as an hour, in contrast to several weeks using traditional methods, practitioners usually discover that treated patients have fewer side effects such as root canals or tooth sensitivity following this dental treatment.

CAD/CAM technologies are used frequently in the dental laboratory industry, allowing the very accurate fabrication of dental prostheses ranging from orthodontic appliances to dental implants, and from crowns to long-span fixed bridges. The accuracy afforded by this technology in dental laboratory technology is unprecedented.

List of CAD/CAM Dental software products

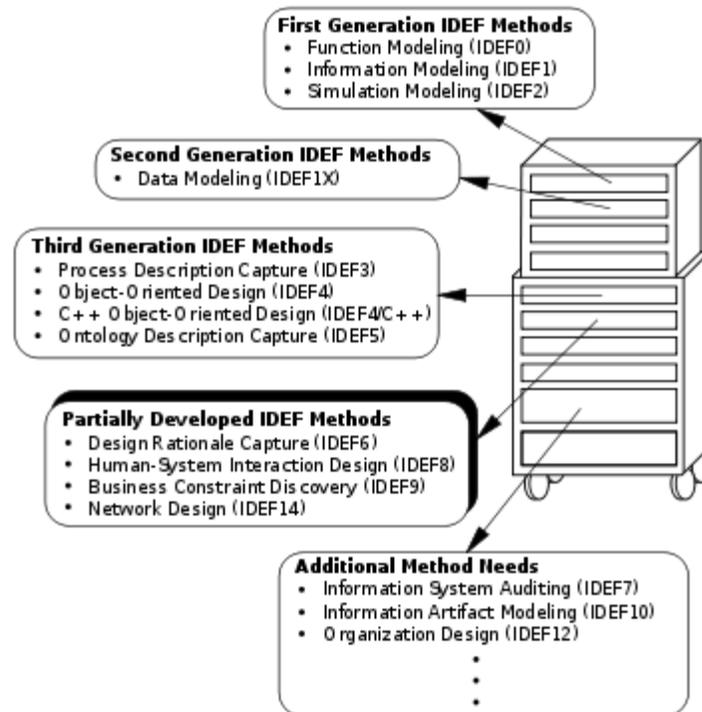
- CEREC, software for manufacturing crowns, veneers, onlays and inlays can be prepared, using different types of ceramic material.
- Delcam dental solutions, for the design and manufacture of copings and bridge frameworks, including full crowns, abutments, dental bars, inlays/onlays and implant bridges
- Renishaw plc
- WorkNC Dental from SESCOI, CAD/CAM for automatic machining of prosthetic appliances, implants, bridges or dental structures

List of Dental scanners

- Solutionix, 3D dental scanners, for the quality measurement of dental models and impressions.
- Straumann CARES Digital Solutions, CS2 Scanner

Integrated Computer-Aided Manufacturing

Integrated Computer-Aided Manufacturing (ICAM) is a US Air Force program to develop tools, techniques, and processes to support manufacturing integration and has influenced the computer-integrated manufacturing (CIM) and computer-aided manufacturing (CAM) project efforts of many companies.



Overview of the IDEF methods developed in the ICAM program.

The ICAM program was founded in 1976 and initiative managed by the US Air Force at Wright-Patterson as a part of their technology modernization efforts. The program initiated the development a series of standard for modeling and analysis in management and business improvement, called *Integrated Definitions*, short IDEFs.

Overview

The USAF ICAM program was founded in 1976 at the US Air Force Materials Laboratory, Wright-Patterson Air Force Base in Ohio by Dennis E. Wisnosky and Dan L. Shunk and others. In the mid-1970s Joseph Harrington had assisted Wisnosky and Shunk in designing the ICAM program and had broadened the concept of CIM to include the entire manufacturing company. Harrington considered manufacturing a "monolithic function".

The ICAM program was visionary in showing that a new approach was necessary to achieve integration in manufacturing firms. Wisnosky and Shunk developed a "wheel" to

illustrate the architecture of their ICAM project and to show the various elements that had to work together. Wisnosky and Shunk were among the first to understand the web of interdependencies needed for integration. Their work represents the first major step in shifting the focus of manufacturing from a series of sequential operations to parallel processing.

The ICAM program has spent over \$100 million to develop tools, techniques, and processes to support manufacturing integration and has influenced the CIM project efforts of many companies. The Air Force's ICAM program recognizes the role of data as central to any integration effort. *Data is to be common and shareable across functions.* The concept still remains ahead of its time, because most major companies will not seriously begin to attack the data architecture challenge until well into the 1990s. The ICAM program also recognizes the need for ways to analyze and document the major activities performed within the manufacturing establishment. Thus, from ICAM came the IDEFs, the standard for modeling and analysis in management and business improvement efforts. IDEF means ICAM DEFinition.

The impact

Standard data models

In order to get real meaning out of the data, we must also have formulated, and agreed on, a model of the world the data describes. We now understand that this actually involves two different kinds of model:

- the static associations between the data and the real-world physical and conceptual objects it describes, called the information model, and
- the rules for the use and modification of the data, which are derived from the dynamic characteristics of the objects themselves, called the functional model.

The significance of these models to data interchange for manufacturing and materials flow was recognized early in the Air Force Integrated Computer Aided Manufacturing (ICAM) Project and gave rise to the IDEF formal modeling project. IDEF produced a specification for a formal functional modeling approach (IDEF0) and an information modeling language (IDEF1). The more recent "Product Data Exchange Specification" (PDES) project in the U.S., the related ISO Standard for the exchange of product model data (STEP) and the Computer Integrated Manufacture Open Systems Architecture (CIMOSA) [ISO87] project in the European Economic Community have whole heartedly accepted the notion that useful data sharing is not possible without formal semantic data models of the context the data describes.

Within their respective spectra of efforts, each of these projects has a panoply of information models for manufactured objects, materials and product characteristics, and for manufacturing and assembly processes. Each also has a commitment to detailed functional models of the various phases of product life cycle. The object of all of these

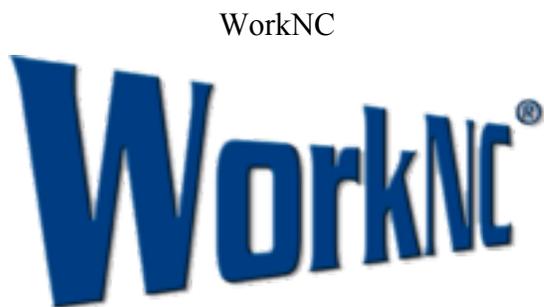
recent efforts is to standardize the interchange of information in many aspects of product design, manufacture, delivery and support.

Further research with ICAM Definitions

The research in expanding and applying the ICAM Definitions have proceeded. In the 1990s for example the Material Handling Research Center (MHRC) of the Georgia Institute of Technology and University of Arkansas had included the it in their Information Systems research area. That area focuses on the information that must accompany material movements and the application of artificial intelligence to material handling problems. MHRC's research involves expanding the integrated computer-aided manufacturing definition (IDEF) approach to include the information flow as well as the material flow needed to support a manufacturing enterprise, as well as models to handle unscheduled events such as machine breakdowns or material shortages. Past research resulted in software to automatically palletize random-size packages, a system to automatically load and unload truck trailers, and an integrated production control system to fabricate optical fibers.

Chapter-9

WorkNC



| | |
|---------------------------|---|
| Original author(s) | Sescoi |
| Initial release | 1988 |
| Stable release | G3 V21 |
| Operating system | Windows |
| Available in | English, French, German, Spanish, Portuguese, Italian, Czech, Dutch, Turkish, Russian, Japanese, Chinese, Korean |
| Type | CAM |

WorkNC is a Computer aided manufacturing(CAM) software developed by SESCOI for 2, 2.5, 3, 3+2 and 5-axis machining. WorkNC is used by more than 25% of companies in demanding countries such as Japan and is well known for having always focused on automation and ease of use since its first release in 1988. WorkNC-CAD was introduced in 2002, making WorkNC a complete CAD/CAM product, one of the worldwide market leaders in its field. The typical users of WorkNC belong to the following industries: automotive, aerospace and defense, engineering, medical & dental, tooling, mold and die manufacturing.

History



Audi concept car, machined using **WorkNC**.

The first version of WorkNC CAM software was released by Sescoi in 1988. The driving forces behind the product were Bruno Marko, president of Sescoi, and Gerard Billard, R&D Innovation Manager.

In the late eighties, CNC programming of complex components was a difficult and lengthy process. This was when Sescoi identified the need for 3-axis CAM software and pioneered the development of WorkNC, a new, reliable and automatic 3-axis CAM system. Over the years, Sescoi has stuck to the original philosophy behind WorkNC : to speed up toolpath calculations, ensure optimum reliability to facilitate machining directly into hard materials, and maximize automation and ease of use so that programming can be carried out on the shop floor. Automation is a theme which runs throughout the development history of the software. According to Bruno Marko, “WorkNC’s aim has always been to be the closest thing possible to a ‘single-button’ CAM”.

Salomon Group was the first customer to use WorkNC in 1988 and has now been using it for over 20 years to manufacture ski boots and other sports equipment. Demand for WorkNC took off and Sescoi expanded its operations by opening offices in the USA in 1991, Germany and Japan in 1995, UK in 1997, followed by Spain, India, China and Korea.

The company released WorkNC-CAD in 2002 and WorkNC 5-axis in 2003.

WorkNC G3, the third generation of the software with an integrated and intuitive CAD/CAM interface, was released in 2007.

In 2008 Sescoi launched WorkXPlore 3D, a high-speed collaborative viewer for analyzing and sharing 3D CAD files without having to rely on the original CAD application.

In 2009 the company launched WorkNC Dental, a CAD/CAM software for automatic 3 to 5-axis machining of prosthetic appliances, implants or dental structures, and also launched WorkNC Wire EDM, a software for Wire EDM. In 2010 Sescoi launched WorkNC-CAD Hybrid Modeling, a powerful 3D CAD software for 3D model design, reparation and machining preparation, with new surface design capabilities tightly integrated with solid modeling functionality, powered by D-Cubed software components.

WorkNC is supported from Sescoi offices in the USA, UK, France, Germany, Spain, Japan, India, China and Korea, and more than 50 distributors around the world.

Main functionality

WorkNC CAM main functions include:

- Automatic geometry and machining zone detection and management
- Specific fluid and progressive toolpaths designed for High Speed Machining
- Full user stock definition (block, CAD, STL)
- Dynamic 3 and 3+2 stock Management (Real time toolpath updated)
- Complete tool and holder collision check with automatic stock update
- Powerful toolpath editor
- Virtual 3D machine representation and machining simulation (dynamic editing of points and vectors)
- Powerful tool and holder library (holder components managed)
- Automatic HTML workshop documentation.
- Fore-casted calculation and machining times can be exported to WorkPLAN, the ERP software from Sescoi
- User predefined machining sequences for automatic machining
- Machining from STL files and point clouds
- Batch mode calculations
- Comprehensive postprocessor generator (NURBS, cycles, circular interpolation...)

Roughing toolpaths

- Global Roughing and Re-roughing toolpath designed and optimized for HSM (high speed machining)
- A range of specific toolpaths with trochoidal, spiral or plunging movement, including HVR (high volume roughing)

- Roughing strategies use the Tool and Holder collision check with an automatic update of the stock
- Automatic calculation and machining of rest areas based on dynamic stock
- Re-machining toolpaths enable automatic rest material machining with increasingly smaller tools

Finishing toolpaths

- A wide range of Finishing toolpaths optimized for HSM machining
- Z Level finishing, Planar finishing, Flat surface finishing, Contour finishing, Edge finishing
- Automatic Rest-material finishing with a sequence of progressively smaller tools
- 3D Display of rest-material areas
- Automatic 5 Axis conversion possible

2 and 2.5 Axis toolpaths

Range of 2 and 2½-axis strategies include:

- Pocketing, Contouring, Curve machining, Engraving, Rib machining, Facing, Drilling, Tapping ...
- Automatic Drilling Module
- Automatic feature detection and recognition, Pre-defined drilling sequence selection, Automatically generated drilling operations, Deep hole and intersecting hole drilling management
- Customized Postprocessor

5 Axis toolpaths

- Automatic 3 to 5-axis conversion with WorkNC Auto 5
- Wide range of Simultaneous 4 and 5 Axis toolpaths
- 5 Axis Rolling, Planar finishing, Spiral Blade, Impeller, Tube, Laser ...
- Collisions detection and machine limits management.

CAD formats imported

WorkNC can read the following CAD file formats: DXF, STEP, IGES, CATIA V4 & V5, Unigraphics, SolidWorks, SolidEdge, Pro/E, Parasolid, STL, etc.

Complementary products

WorkNC Dental



Chrome-cobalt disc with bridges and crowns manufactured using WorkNC Dental CAD/CAM

WorkNC Dental is the CAD/CAM software from Sescoi for automatic 3 to 5-axis machining of prosthetic appliances, implants, crowns, bridge implants and dental structures.

WorkNC Dental has an easy to use graphic user interface designed for labs and dental technicians who are not machining experts. It is the closest thing to a « one button CAM » for the dental sector and has been developed and tested in collaboration with dental professionals and professional bodies. Its automatic 3 and 5-axis toolpaths combine with technology for nesting and orientating prosthetic implants. Additionally, tools for adding support pins and identification markings make the software an ideal solution for rapid prosthesis manufacture.

STL and native dental CAD formats are imported into WorkNC Dental ready for “machining wizards” to guide the user through the manufacturing process. The system automatically selects tools, machining sequences and cutting conditions optimized for particular types of material, such as chrome-cobalt, titanium and zirconia, and for particular prostheses, such as copings and bridges. The intricate nature of some implants

requires the use of 5-axis machining methods to reach every part of the job. Traditionally requiring highly skilled programmers, WorkNC Dental overcomes this problem with its automated 5-axis software. Intelligence within the system considers the limitations and kinematics of the machine tool itself to automatically produce reliable and collision free 5-axis toolpaths.

WorkNC MPM (Multi-Part Machining)

WorkNC MPM is a CAD/CAM module that allows users to easily machine multiple parts simultaneously on the same machine. Operators are confronted by a number of problems as a result of multiple tool changes (collision management, down time and lack of precision).

WorkNC's MPM module eliminates these problems and offers productivity gains by optimizing the machining operations. No programming is required and no supplementary part origin data need to be defined.

There is also a significant reduction in the number of tool changes required. Problems due to incorrect tool numbering are eliminated, and there are no collisions between the parts.

The use of predefined pallet systems ensures rapid and safe machining.

MPM can consider a complete pallet load of electrodes as one part, using each tool across all the electrodes on the pallet. The software ensures collision-free cutter paths as it moves across the job, and eliminates the many tool change operations that would otherwise be required. MPM automatically determines a safe retract height above the tallest electrode, and the clear graphics enable the programmer to identify the slightest potential problem.

The main benefits of WorkNC MPM are: No programming is necessary on the machine, Simple and reliable part positioning, Unsupervised machining, Optimal use of clamping assemblies and palettes, Significant reduction in number of tool changes, Flexibility and adaptability during modifications, Part symmetry and rotation without having to recalculate toolpaths.

WorkNC LMP (Layer Milling Process)

WorkNC LMP is a CAD/CAM software for cutting parts in layers, a well-known technique for machining deep and narrow cavities.

WorkNC-LMP automatically divides 3D models and creates roughing and finishing toolpaths for each layer. This technology can be used on any machining center, simplifying the programming and cutting of complex shapes by building them up in manageable sections.

Sescoi worked in collaboration with F. Zimmermann, combining the speed and accuracy of the dedicated Zimmermann LMC (Layer Milling Centre) with SESCOI's WorkNC-LMP. By using the software in conjunction with the machine all the toolpaths and special machine control sequences are automatically generated to cut parts in unmanned operation.

The LMC works from underneath, using high-speed techniques to machine each layer in turn. As each layer is roughed and finished a new plate is bonded, ready for the next machining operation, this continues until the finished part is built up. WorkNC-LMP automates this process starting from the CAD model, firstly splitting it into layers, and then generating both the roughing and finishing toolpaths for each layer. This technique is ideal for high speed machining enabling the use of short and rigid cutters, and eliminates the possibility of a collision. WorkNC-LMP allows to select the material from a materials library and controls factors such as surface roughening prior to bonding, cutting adhesive channels to control excess glue, overlapping of cutter paths to remove traces of glue, and paths for the application of adhesive between layers. It provides the visual control of all toolpaths and the calculation of the estimated total machining time.

WorkNC-LMP combines the advantages of generative rapid prototyping with conventional machining, and is the latest example of the company's resolve to make new processes a reality for its customers.

WorkNC Wire EDM

WorkNC Wire EDM is a CAD/CAM software for Wire Electrical discharge machining. Dialog boxes guide the user through the system. Functions within WorkNC Wire EDM allow the extraction of cross sections, ready for 2- or 4-axis cutting. Alternatively, the 3D surfaces of the CAD model can be used directly.

It also includes graphical verification to automatically check for collisions and the maximum wire angle possible on each individual EDM machine. The latest version makes it easy to extract and link 4-axis wire paths. Dialog boxes guide the user through the process making it simple to add tags, create roughing and finishing wire paths, lead in and out moves, a range of corner strategies, and tag removal cycles. Postprocessors and technology libraries are included for all leading machines.

WorkNC-CAD

WorkNC-CAD is a manufacturing CAD software with surface and solid modeling functions. It is included free as a standard integrated component of WorkNC. It provides features required to design and manufacture molds, dies, and tooling without the need for additional software applications or outsourcing.

WorkNC-CAD has advanced intelligent surface morphing for filling simple or complex cavities, automatic 2D feature recognition and cycle definition for drilling, counterboring, reaming and tapping, as well as automatic mold and die core separation.

WorkNC-CAD Hybrid Modeling

WorkNC-CAD Hybrid Modeling (HM) is a 3D CAD software launched by Sescoi in 2010 for 3D model design, reparation and machining preparation, with surface design capabilities integrated with solid modeling functionality in a user friendly environment. It can be used as an independent CAD product. It is powered by D-Cubed software components from Siemens PLM.

WorkNC-CAD HM works on solid and surface models making use of parametric commands to easily manipulate and repair CAD data. Along with multiple CAD translators, it includes modules for electrode creation, core/cavity separation and Wire EDM. The WorkNC Electrode module makes use of the WorkNC-CAD Hybrid Modeling capabilities to extract electrode shapes directly from solid or surface models. The electrode model can be modified and extended, and tool holders added from a library to produce a complete electrode. WorkNC's collision checking ensures the electrode does not collide with any surrounding surfaces, automatically adding extensions as required. Documentation and electrode coordinate systems are produced by the software to ensure correct positioning for the EDM operations.

WorkNC-CAD Hybrid Modeling features include:

- Large visualization area
- Construction tree that enables simplified CAD/CAM entity management
- Context menus accessible in the construction tree and in the graphic window
- Access to features according to context
- Real-time preview of surface and solid modeling functions
- Configurable interface (keyboard shortcuts, position of adjustable toolbars).

When used in its standalone design version by toolmakers in their technical departments, it offers mold, tool and die businesses a uniform CAD product throughout the entire manufacturing process.

WorkXPlore 3D

WorkXPlore 3D is a high-speed collaborative viewer for analyzing and sharing 3D CAD files without having to rely on the original CAD application. The software is very easy to use and is designed for users who are not CAD experts to explore 2D/3D CAD files of multiple types. WorkXPlore enables the user to measure 3D parts and offers advanced analysis features allowing the user to determine undercut areas, plane surfaces, thickness, volumes, surfaces, weight and also to perform dynamic cross-sectional visualization. 2D drawings are no longer necessary as it is possible to directly add dimensional and geometric measurements, annotations and labels to the 3D model.

WorkXPlore 3D also allows the user to send 3D parts and assembly files to sub-contractors, customers or co-workers via a standalone and extremely compact application which can be easily transmitted via the Internet as an executable file. The recipient can

immediately display and work on the 3D model received. WorkXPlore 3D can open and process large 3D files extremely quickly. A free viewer version and a free evaluation version are available.

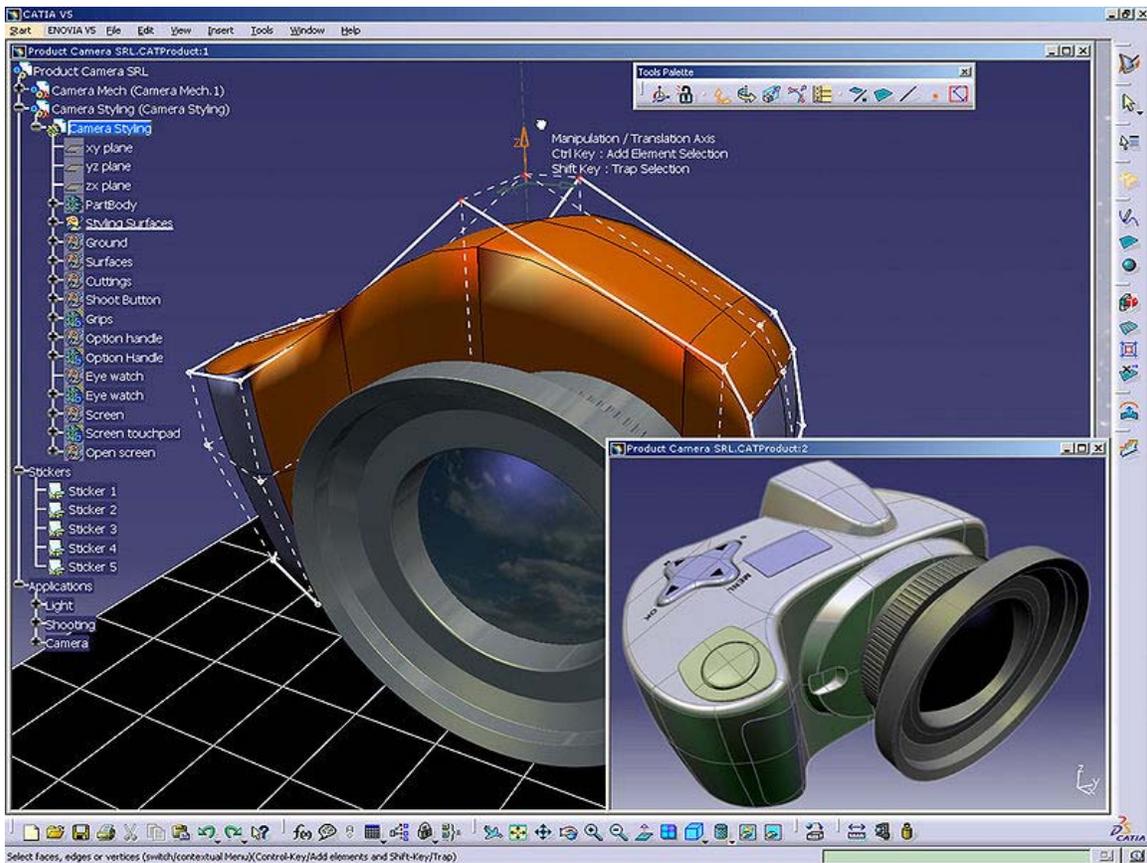
Chapter-10

CATIA and HyperSizer

CATIA



North Leg of the Eiffel Tower



Concept Product

CATIA (Computer Aided Three-dimensional Interactive Application) is a multi-platform CAD/CAM/CAE commercial software suite developed by the French company Dassault Systemes and marketed worldwide by IBM. Written in the C++ programming language, CATIA is the cornerstone of the Dassault Systemes product lifecycle management software suite.

The software was created in the late 1970s and early 1980s to develop Dassault's Mirage fighter jet, then was adopted in the aerospace, automotive, shipbuilding, and other industries.

CATIA competes in the CAD/CAM/CAE market with Siemens NX, Pro/ENGINEER, Autodesk Inventor, and SolidEdge.

History

CATIA started as an in-house development in 1977 by French aircraft manufacturer Avions Marcel Dassault, at that time customer of the CADAM CAD software.

Initially named **CATI** (*Conception Assistée Tridimensionnelle Interactive* — French for *Interactive Aided Three-dimensional Design*) — it was renamed CATIA in 1981, when Dassault created a subsidiary to develop and sell the software, and signed a non-exclusive distribution agreement with IBM.

In 1984, the Boeing Company chose CATIA as its main 3D CAD tool, becoming its largest customer.

In 1988, CATIA version 3 was ported from mainframe computers to UNIX.

In 1990, General Dynamics Electric Boat Corp chose CATIA as its main 3D CAD tool, to design the U.S. Navy's Virginia class submarine.

In 1992, CADAM was purchased from IpM and the next year **CATIA CADAM V4** was published. In 1996, it was ported from one to four Unix operating systems, including IBM AIX, Silicon Graphics IRIX, Sun Microsystems SunOS and Hewlett-Packard HP-UX.

In 1998, an entirely rewritten version of CATIA, **CATIA V5** was released, with support for UNIX, Windows NT and Windows XP since 2001.

In 2008, Dassault announced and released **CATIA V6**. While the server can run on Microsoft Windows, Linux or AIX, client support for any operating system other than Microsoft Windows is dropped.

Release history

| Name/Version | Latest Build Number | Original Release Date |
|---------------------|----------------------------|------------------------------|
| CATIA v4 | R17 | 1993 |
| CATIA v5 | R20 | 1998 |
| CATIA v6 | R2010 | 5/29/2008 |
| CATIA v6 | R2011x | 11/23/2010 |

Features

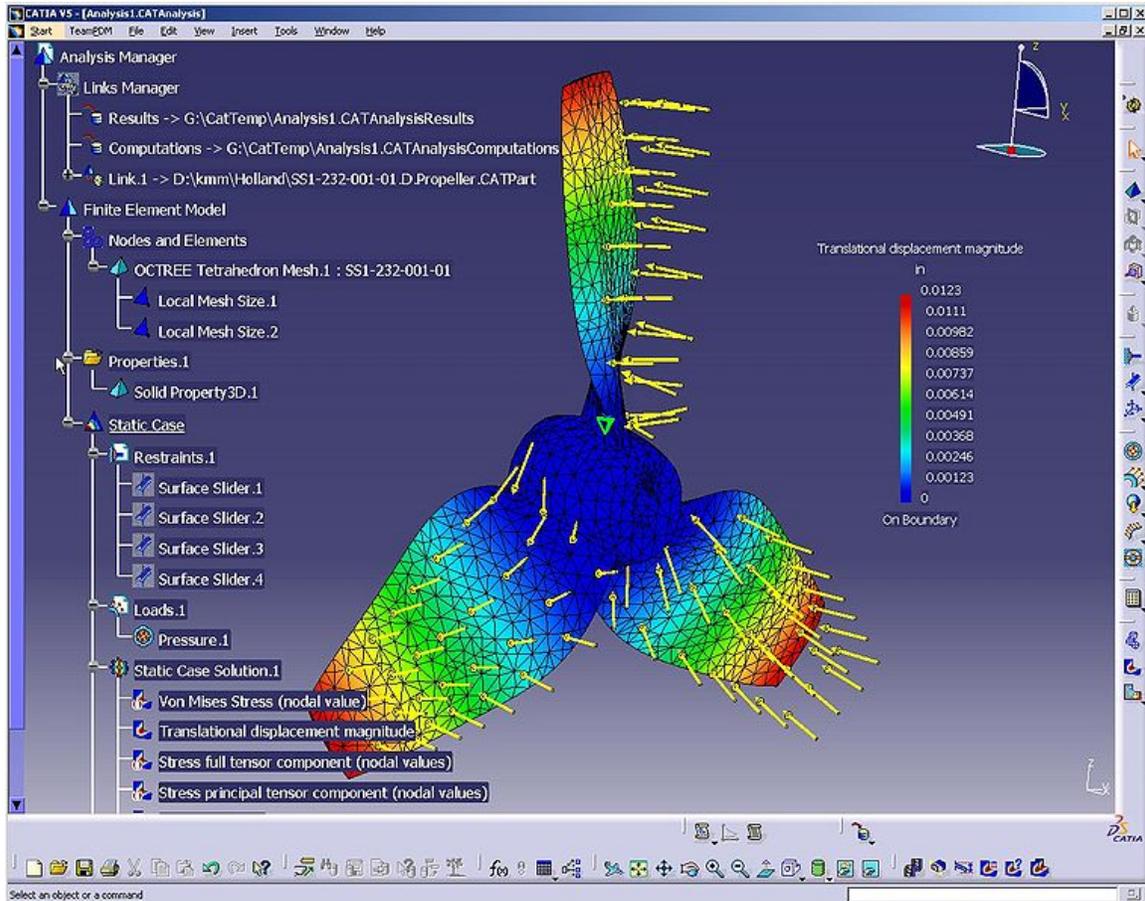
Commonly referred to as a 3D Product Lifecycle Management software suite, CATIA supports multiple stages of product development (CAx), from conceptualization, design (CAD), manufacturing (CAM), and engineering (CAE).

CATIA can be customized via application programming interfaces (API). V4 can be adapted in the Fortran and C programming languages under an API called CAA (Component Application Architecture). V5 can be adapted via the Visual Basic and C++ programming languages, an API called CAA2 or CAA V5 that is a component object model (COM)-like interface.

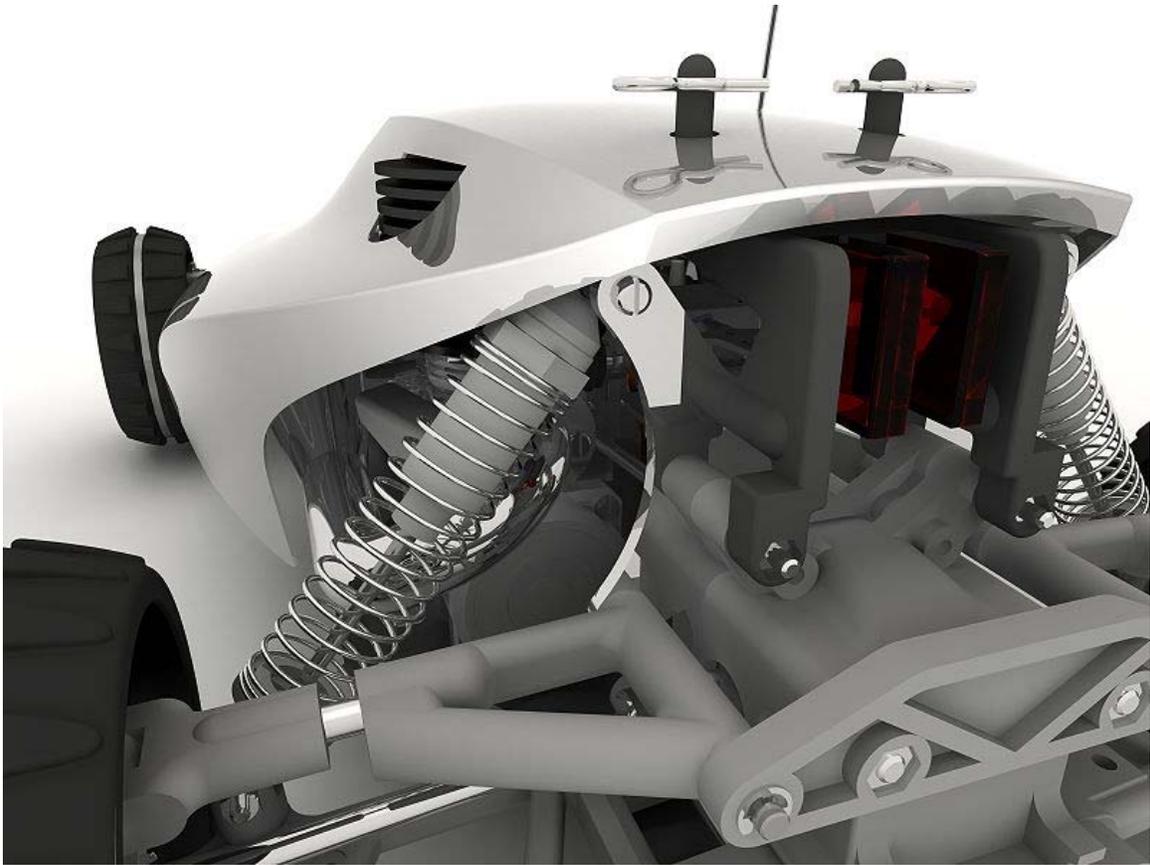
Although later versions of CATIA V4 implemented NURBS, V4 principally used piecewise polynomial surfaces. CATIA V4 uses a non-manifold solid engine.

Catia V5 features a parametric solid/surface-based package which uses NURBS as the core surface representation and has several workbenches that provide KBE support.

V5 can work with other applications, including Enovia, Smarteam, and various CAE Analysis applications.



FEM & Structural Analysis



CATIA Rendering

Supported operating systems and platforms

CATIA V6 runs only on Microsoft Windows And MAC with limited products.

CATIA V5 runs on Microsoft Windows (both 32-bit and 64-bit), and as of *Release 18 Service Pack 4* on Windows Vista 64. IBM AIX, Hewlett Packard HP-UX and Sun Microsystems Solaris are supported.

CATIA V4 is supported for those Unixes and IBM MVS and VM/CMS mainframe platforms up to release 1.7.

CATIA V3 and earlier run on the mainframe platforms.

Notable industries using CATIA

CATIA is widely used throughout the engineering industry, especially in the automotive and aerospace sectors. CATIA V4, CATIA V5, Pro/ENGINEER, NX (formerly Unigraphics), and SolidWorks are the dominant systems.

Aerospace

The Boeing Company used CATIA V3 to develop its 777 airliner, and is currently using CATIA V5 for the 787 series aircraft. They have employed the full range of Dassault Systemes' 3D PLM products — CATIA, DELMIA, and ENOVIA LCA — supplemented by Boeing developed applications.

The development of the Indian Light Combat Aircraft has been using CATIA V5.

Chinese Xian JH-7A is the first aircraft developed by CATIA V5, when the design was completed on September 26, 2000.

European aerospace giant Airbus has been using CATIA since 2001.

Canadian aircraft maker Bombardier Aerospace has done all of its aircraft design on CATIA.

The Brazilian aircraft company, EMBRAER, use Catia V4 and V5 to build all airplanes.

Vought Aircraft Industries use CATIA V4 and V5 to produce its parts.

The British Helicopter company, Westlands, use CATIA V4 and V5 to produce all their aircraft. Westlands is now part of an Italian company called Finmeccanica the joined company calls themselves AgustaWestland.

The main supplier of helicopters to the U.S Military forces, Sikorsky Aircraft Corp., uses CATIA as well.

Automotive

Many automotive companies use CATIA to varying degrees, including BMW, Porsche, Daimler AG, Chrysler, Honda, Audi, Volkswagen, Bentley Motors Limited, Volvo, Fiat, Benteler AG, PSA Peugeot Citroën, Renault, Toyota, Ford, Scania, Hyundai, Škoda Auto, Tesla Motors, Valmet Automotive, Proton, Tata motors and Mahindra & Mahindra Limited. Goodyear uses it in making tires for automotive and aerospace and also uses a customized CATIA for its design and development. Many automotive companies use CATIA for car structures — door beams, IP supports, bumper beams, roof rails, side rails, body components — because CATIA is very good in surface creation and Computer representation of surfaces.

Shipbuilding

Dassault Systems has begun serving shipbuilders with CATIA V5 release 8, which includes special features useful to shipbuilders. GD Electric Boat used CATIA to design the latest fast attack submarine class for the United States Navy, the *Virginia* class.

Northrop Grumman Newport News also used CATIA to design the *Gerald R. Ford* class of supercarriers for the US Navy.

Transportation

Bombardier Transportation, Canada is using CATIA software to design its entire fleet of Train engines, Coaches.

Other

Architect Frank Gehry has used the software, through the C-Cubed Virtual Architecture company, now Virtual Build Team, to design his award-winning curvilinear buildings. His technology arm, Gehry Technologies, has been developing software based on CATIA V5 named Digital Project. Digital Project has been used to design buildings and has successfully completed a handful of projects.

File compatibility and CATIA V4 conversion

Dassault Systemes provides utilities to convert CATIA V4 data files so they are accessible to CATIA V5 and CATIA V6. Still, cases show that there can be issues in the data conversion from CATIA V4 to V5, from either differences in the geometric kernel between CATIA V4 and CATIA V5, or by the modelling methods employed by end users. Experiment results show that there can be data loss during the conversion (from 0% to 90%). The percentage loss can be minimized by using the appropriate pre-conversion clean-up, choosing the appropriate conversion options, and clean-up activities after conversion.

Third-party file translators such as TransMagic also up-convert CATIA V4 files and also expand the number of non-CATIA file types that can natively opened and saved.



CATIA Rendering of a diamond

Future implementations

Dassault Systemes released CATIA Version 6 (V6) in mid-2008. In mid-2010, Dassault Systemes launched Catia V6 R2011, the latest release of its PLM2.0 platform while still continuing to support and improve its Catia V5 software. Latest release of Catia V5 is R20 SP5.

HyperSizer



HyperSizer Software

Developers: Collier Research Corp.

Latest Release: 5.8.11

Operating System: Windows XP, Windows Vista

HyperSizer Composite & Laminate Analysis Software

HyperSizer is a composite material analysis software providing stress analysis and preliminary structural design for weight reduction, system level trade studies, and part sizing including sizing stiffened panel cross-sectional dimensions, material selection, and optimum composite layup stacking. HyperSizer performs detailed failure analyses and reporting as a post process to a loads finite element model (FEM).

It is an enterprise tool used by both designers and stress analysts from early concept to final design.

For over a decade, HyperSizer has significantly reduced weight on major aerospace programs with rapid analysis and composite optimization. Originally developed at NASA, it is now greatly enhanced and supported as a commercial product worldwide. HyperSizer is not CAD and is not finite element analysis (FEA).

Software

HyperSizer is a software system for management of all data associated with the structural analysis and test data of a major aircraft program. Multiple databases can be setup, with each database able to contain hundreds of variations of an airframe configuration (i.e. different FEMs), material properties, panel and beam concepts, dimensions, and loads. This approach provides apples-to-apples weight prediction comparisons and a guaranteed store of all margins of safety for every configuration. The database also provides an organized and efficient means to capture essential data related to a project with a guaranteed ability to immediately locate and retrieve historical data. Stress reports can be generated at the click of a button at any time documenting current status of a project, indicating critical margins, critical load cases, and critical structural parts.

Analyses

HyperSizer performs hundreds of different analyses such as panel buckling, crippling, beam-column, bonded and bolted joint, composite strength to damage initiation, and damage tolerance criteria, etc. for the entire vehicle from engine nacelles to airframe surface panels and substructure. HyperSizer imports a FEM and manages all data associated with a configuration. Wing spars and ribs can consider a range of materials and panel concepts that are different than the subset of user-determined design options for the wing skins and fuselage body. HyperSizer also analyzes and optimizes internal beams such as spar caps and many other open and closed shapes. A primary foundational

capability of HyperSizer is to accurately analyze any panel concept without the need to discretely mesh with finite elements the shape of the stiffeners or their spacing. This permits tremendous flexibility and rapid turnaround of trades with different panel concepts all from the same coarsely meshed FEM.

Progressive Design Process

The HyperSizer Progressive Design Process consists of three activities; all three activities can interact with each other throughout design maturation. HyperSizer provides unique automation and integration capabilities to each of these design activities.

First, there is a funneling process performed in stages to target an optimum design. Innovative “back to the drawing board” concepts are proposed, evaluated, and filtered out for the next stage of the design maturation process.

Second, an incremental process of including more computationally demanding analysis solutions starting with damage initiation, tracking the progression of failure, and ending with the resulting residual strength at ultimate failure.

Third, an incremental process of including more design detail, such as bonded and bolted joints, ply drop-offs, etc. for both optimization and analysis.

Some Projects & Users

NASA

Langley Research Center (Hampton, VA) Marshall Space Flight Center (Huntsville, AL)
Glenn Research Center (Cleveland, OH) Johnson Space Center (Houston, TX) Ames
Research Center (Moffett Field, CA)

Boeing

Huntington Beach, CA (Integrated Defense & Space) Seattle, WA (Commercial
Airplanes) Seattle, WA (Integrated Defense Systems) Phantom Works (Aerospace R&D)
Houston, TX (NASA Systems)

Lockheed Martin

Denver, CO (Space Systems) Marietta, GA (Aeronautics) New Orleans, LA (Michoud
Space Systems) Palmdale, CA (Aeronautics Skunk Works) Fort Worth, TX (Aeronautics)
Syracuse, NY (Maritime Systems & Sensors)

Northrop Grumman

El Segundo, CA (Integrated Systems) San Diego, CA (Teledyne Ryan) Biloxi, MS (Ship
Systems) Newport News, VA (Ship Systems)

Goodrich Aerostructures Chula Vista, CA

AFRL (Air Force Research Lab)

Wright-Patterson AFB, OH Edwards AFB, CA Hill AFB, Ogden, UT Albuquerque, NM

GM - General Motors R & D Center Detroit, MI

Product Modules

The three primary products of the HyperToolbox Composite Software are HyperSizer Material Manager, HyperSizer Basic, and HyperSizer Pro, all designed for Windows XP/2000.

Material Manager

With Material Manager, you can build composite laminates with any arbitrary stacking of material forms or material types. Use native Windows cut, paste, and copy functions for quick ply insertions and layup arrangements. Define and save stack templates and use other advanced layup tools. Graph failure envelopes and stress/strain profiles automatically. Perform highly interactive "what-if" design changes and see their effects in real time.

Basic

HyperSizer Basic includes all the functionality of HyperSizer Material Manager, such as composite laminate analysis, plus stress analysis and detailed design optimization for over 50 different stiffened and sandwich panel and beam concepts. Any material or cross-sectional dimension can be optimized to all load cases. Apply general edge loadings and/or boundary conditions through the Free Body Diagram Software tab and solve for the resulting stresses and structural integrity using over 100 different failure analyses. Analyses include traditional industry methods and modern analytical and numerical solutions.

Pro

HyperSizer Pro includes all the functionality of Basic, plus coupling with FEA for system level analysis and automatic finite element model resizing and results display. Using panel, beam, and laminate analyses from the Basic and Material Manager products, Pro provides a complete and detailed analysis of entire systems such as airframes, rocket engines, ship hulls, and train bodies. Complete margin-of-safety (MS) summaries of all potential structural failure modes for all load cases are reported. FEA-computed internal loads are retrieved and input automatically for sizing the panels and beams. HyperSizer then creates generalized thermoelastic stiffness terms to send back to the FEM for another iteration of computed internal load paths. HyperSizer controls this iterative convergence and controls FEM global responses such as wing-twist deflections.

In addition to these three primary products, are two highly integrated tools that share the same database for the analysis, material data input, and computed data output:

HyperSizer TPS

The HyperSizer Thermal Protection System (TPS) capability is available as an add-on

module to HyperSizer Pro. HyperSizer TPS combines the disciplines of TPS analysis with structural analysis in a single program. This integrated TPS-Structures capability provides a commercial quality TPS analysis and sizing program that will become a standard analysis method for future space launch vehicles.

HyperSizer Micromechanics

HyperSizer Micromechanics is an add-on module for HyperSizer Structural Sizing Software and was developed by integrating a powerful University/NASA-developed Micromechanics analysis capability with HyperSizer. This new add-on module combines the constitutive (fiber/matrix) buildup, non-linear analysis of MAC/GMC (Micromechanics Analysis Code with Generalized Method of Cells) with the data integrity and established laminate and material property editing interface of the HyperSizer product.

Software Specifications

To run on a PC, the operating system must be either Windows XP or 2000. It is recommended to use Windows XP with the latest service pack. HyperSizer has been tested successfully on Microsoft Vista and is compatible, although not Vista-certified.

For the HyperSizer Basic or Pro products, they recommend at least a 600 MHz Pentium III with at least 512 Mbytes of memory. The HyperSizer GUI is much more useful running on large monitors (19 or 21 inch). The HyperSizer Material Manager Product will run effectively on almost all PC hardware.

The minimum required resolution setting to fit the HyperSizer GUI onto the desktop is 1024x768 pixels (with small fonts) but we recommend 1280x1024 with large fonts. With resolutions less than 1024x768, the user must scroll to see some of the forms fully. In addition, high color (more than 256 colors) is recommended for viewing the imbedded graphics.

USING EMULATION: HyperSizer has been successfully installed and used on MAC computers. However, due to the operating system emulation, HyperSizer runs much slower on a MAC than on a PC. The slow down will also be apparent running on Unix.

USING TERMINAL SERVICES: HyperSizer may effectively be used from a NT server with your computer as a client.

HyperSizer has been successfully tested to run in other countries such as France, Italy, Japan, and Korea using XP/2000 with the latest service pack. The latest service pack for Windows 2000 can be obtained from the Microsoft website or from the Windows Update feature of Internet Explorer.

Running HyperSizer with your preferred country OS installation (therefore a non-United States installation) is possible and requires only a slight change to your system. As an example, if you are in Italy, we suggest you install the Italian version of XP/2000.

However, before you start HyperSizer, you must make a slight modification to the regional settings of your computer. There are two options. First, you can change the regional settings to "English (United States)" and then HyperSizer will work fine. A preferable option for you may be to choose Italy for your regional settings. In this case, you will need to change the comma to a decimal separator. Open the control panel, and double-click the applet, "Regional Options" ("Opzioni internazionali") and click on the "Numbers" ("Numeri") tab. In the dropdown box labeled, "Decimal Symbol" ("Separatore decimale") change the symbol from a comma (,) to a period (.). Next, in the dropdown list next to "Digit grouping symbol" ("simbolo raggruppamento cifre"), change the symbol from a period (.) to a comma (,) and click the "Apply" ("Applica") button.

Metric and English units are fully supported with version 3.4 and up.

Chapter-11

MPDS4

MPDS, the MEDUSA Plant Design System (since 2006 known as **MPDS4**) is a suite of plant engineering applications for 2D/3D layout, design and modelling of process plants, factories or installations. The system's history is closely tied to the very beginnings of mainstream CAD and the research culture fostered by Cambridge University and the UK government as well as the resulting "Cambridge Phenomenon" MPDS was originally developed for 3D plant design and layout and piping design. Today the software includes modules for 2D/3D factory layout, process and instrumentation diagrams (P&ID), mechanical handling systems design, steel design, ducting (HVAC) design, electrical design and hangers and supports Design. The latest major version, MPDS4 4.0, was released for Microsoft Windows® and Sun® SPARC Solaris in September 2009.

History

MPDS' history is tied in with the Computer-Aided Design Centre (or CADCentre) which was created in Cambridge in 1967 by the UK Government to carry out CAD research.

Famous British computer scientist Dr. Dick Newell worked there on a file-based macro language driven 3D plant design system called PDMS (Plant Design Management System). Together with colleague Tom Sancha he left the CADCentre in 1977 to form a company called Cambridge Interactive Systems or CIS and primarily concentrated on 2D CAD. CIS had developed an electrical cabling solution initially called CABLOS, which was first purchased by Dowty Engineering in about 1979. Another early adopter was BMW, which used the system for car wiring diagrams. CABLOS soon became known and sold as the MEDUSA drafting system under CIS. The proprietary programming language with which MEDUSA version 1 was developed was known as baCIS 1. Around this time, the company also began developing its own 3D modelling kernel for MEDUSA.

Around 1980, CIS partnered with Prime Computer, a U.S.-based computer hardware provider. Prime had an option on the MEDUSA source code should CIS ever fail. In 1983 the U.S. CAD company Computervision purchased CIS.

Computervision/CIS started developing the MEDUSA Plant Design System (MPDS), the first plant design software based on a relational database. Developers knew from their prior experience with a file-based macro-language driven system that the next generation plant design system had to be built on a relational database and with a much more powerful programming language to handle large data volumes, complexity and relationships. Whereas mechanical CAD engineers were developing machinery with a few hundred or maybe a thousand components, plant design engineers typically needed to deal with hundreds of thousands of components. In order to facilitate this work, the baCIS 2 interpretive language and the MDB relational database were developed for MPDS. Existing MEDUSA technology was used to create the 2D and 3D geometric data required for plant layouts. The creation of this data-centric concept separated the 3D visualisation of a plant from the underlying database and allowed engineers to plan and design installations with very large volumes of data, and produce all the required 2D drawings from 3D plant designs.

The first MPDS sales date to around 1988 to NEI Parsons (Northern Engineering Industries later became part of the Rolls Royce Industrial Power Group). Courtaulds Engineering, which had been using MEDUSA since 1983, was also an early MPDS adopter.

In the same year Prime Computer merged with Computervision and adopted the name Computervision in order to concentrate on software, due to declining hardware sales. MEDUSA continued to be developed throughout the 1990s in Cambridge, UK at Computervision's R&D centre at Harston Mill.

In 1993, the next generation of MEDUSA and MPDS was released. What would have been version 13 was released as MEDUSA NG and MPDS NG. They signified the shift from tablet-driven menus to a graphical user interface, although tablets could still be used on that release.

In 1994 Computervision closed its R&D facility in Cambridge, moving to Boston, Massachusetts. As a result, five former Computervision staff members and MEDUSA experts formed the company Quintic Ltd in Cambridge, which continued to provide MEDUSA and MPDS development and consultancy services to Computervision and the MEDUSA customer base. Work included the porting of MEDUSA NG to Microsoft Windows.

In 1998 the American CAD company Parametric Technology Corporation (PTC) acquired Computervision. The development partnership between Quintic and Computervision transferred to PTC.

One of the largest MEDUSA user bases was in the heavily manufacturing-driven economy of Germany. CAD Schroer, a company founded in 1986 by Michael Schroer as a provider of CAD-based engineering services, became a MEDUSA vendor in 1988, having used the software extensively on client projects. The company, which also

provided add-on modules and customisations, had established a development relationship first with Computervision, then with PTC.

In 2001, CAD Schroer acquired all rights to MEDUSA and MPDS from PTC. The development partnership between Quintic and CAD Schroer strengthened, as the two companies worked to create a Fourth Generation release of MEDUSA and MPDS. This included a complete overhaul of the functionality; the development of a graphical user interface (GUI) based on the Qt (framework) technology, the development of data exchange mechanisms and interfaces with third party systems, and the porting to the Linux open source operating system.

In 2005, CAD Schroer acquired its development partner Quintic Ltd, gaining CAD development expertise that dates back to the days of CIS and Prime. CAD Schroer UK remains a software development centre in Cambridge, whose staff continue to develop and support MEDUSA4 and MPDS4 in partnership with CAD development experts at CAD Schroer GmbH in Moers, Germany.

In 2006, CAD Schroer released the Fourth Generation of the MPDS plant design system, MPDS4. Since then the company has continued to develop and extend the functionality of the product suite, including the development of a factory layout module for designing 3D factories based on 2D drawings.

Technical Description

Database Architecture

Multi-user engineering design in MPDS is relational database-driven. The project database can be deployed as a central design database or as a project-specific database and contains component catalogs with assigned component attributes. The database drives the design graphics as well as user administration and can be integrated with other database-driven systems, such as Enterprise Resource Planning (ERP) system.

3D Graphics

MPDS combines use of the HOOPS 3D Graphics System with the relational database, whose catalog component attributes define the visual representation of each component. Because 3D plant models can be generated from catalog-based drawing routines, the demand on computer memory and resources is limited. Plants of hundreds of thousands of components can be designed, edited and exported to a compact .HSF format for external visual review. The system support varying display detail levels, allowing designers to visualise components either in great detail, visually simplified, or merely as an outline object in space, required for effective clash detection.

User Administration

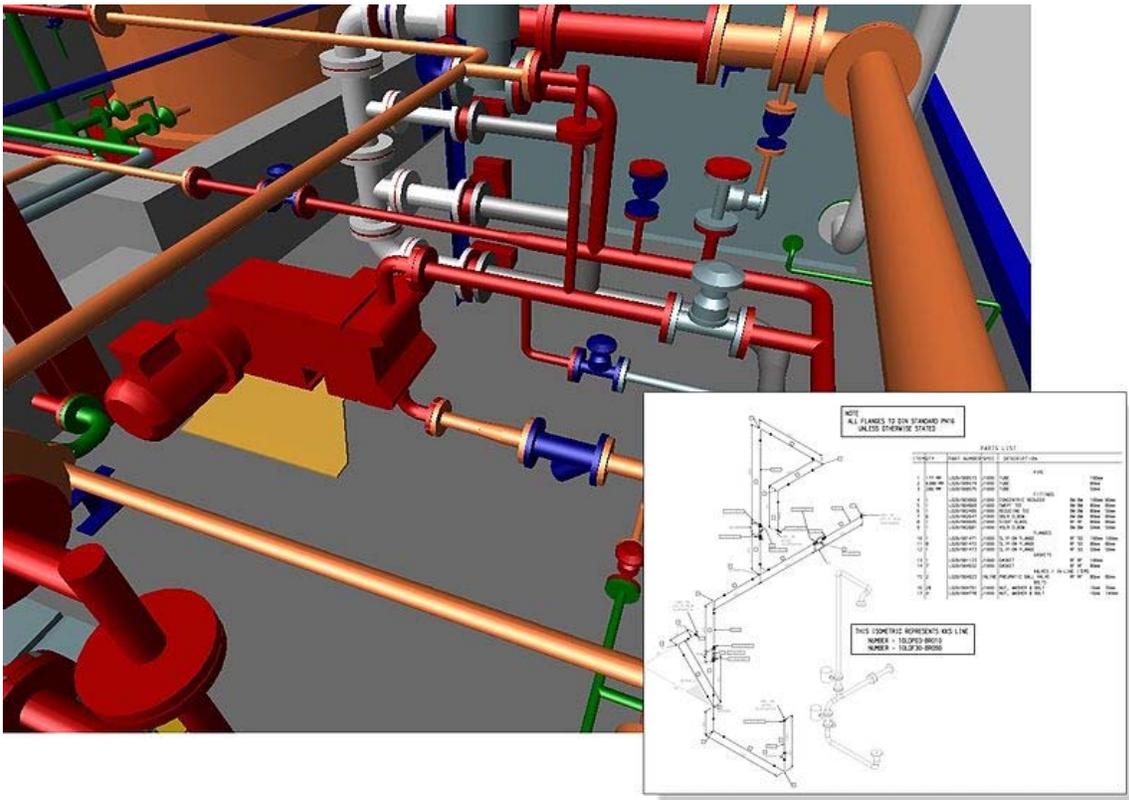
Central user administration and access controls in MPDS allow Administrators to set up a variety of users who can work on a plant design simultaneously, and who can have different access privileges - limited, for example, to certain design disciplines or to certain areas within a plant. This is supported by integrated version and change management.

Quality Assurance

MPDS4 includes hard and soft clash detection, which can be applied to a whole project, to separate systems or between selected components. Consistency checking tools allow users to check work against specific design rules. Results can be passed to customisable reports, and components used in a design are automatically included in parts lists.

Modules

The MPDS4 Assembly Manager is at the core of the plant design and factory layout system and can be extended with several user extensible and customisable modules covering the plant engineering disciplines.

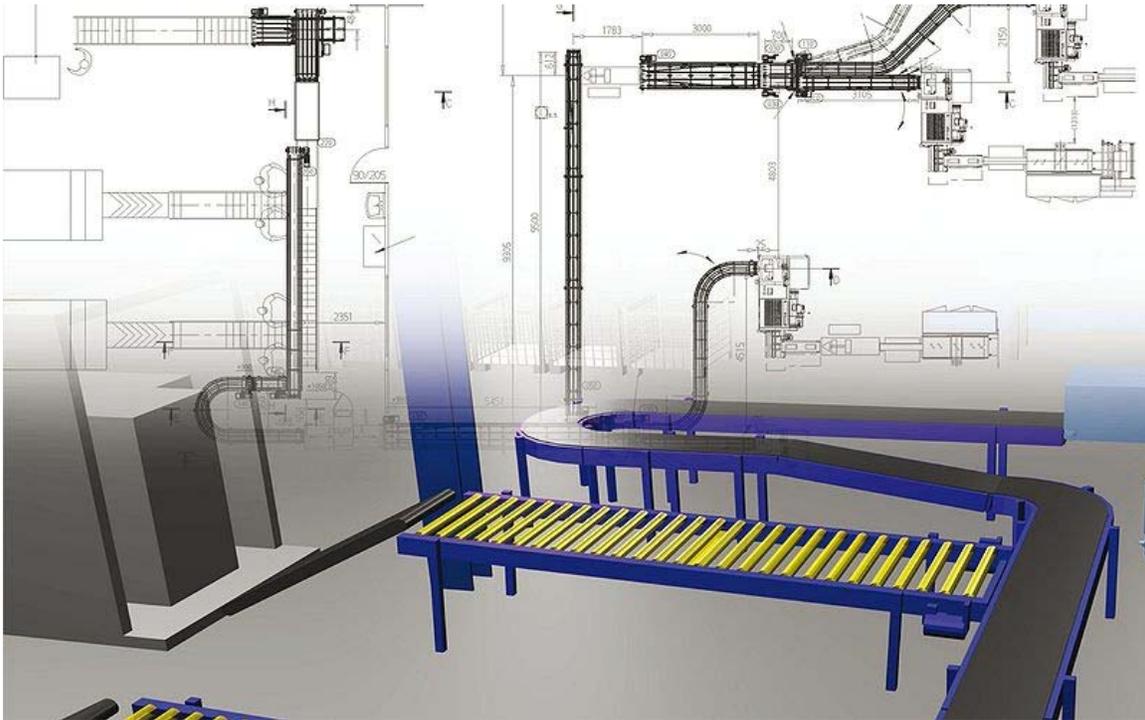


PIPING DESIGN

MPDS4 PIPING DESIGN is an industrial piping design software add-on with extensive libraries of catalogue components to a variety of industrial standards, including DIN, ANSI and BS. Its routing tools are used for loading, positioning and replacing components, manually or automatically. The module supports P&ID-driven piping design and is pipe specification driven, so that only components from the same specification can be connected. MPDS4 PIPING DESIGN is fully integrated with ISOGEN™ (from ALIAS Piping Solutions) for automated piping isometric production.

P&ID

P&ID is an application for creating intelligent process and instrumentation diagrams; for data extraction, and for use of data to create and cross-check the 3D world. Design can be database-driven and based on existing parts lists. P&ID diagrams can be used to form the basis of 2D layouts and 3D designs, with the ability to cross-check P&IDs and automatically load P&ID components not yet included in a 3D plant design in the appropriate position.



MPDS4 Factory Layout offers a hybrid 2D/3D design environment

FACTORY LAYOUT

MPDS4 FACTORY LAYOUT is a hybrid 2D/3D design environment where 2D layouts or drawings are used as the basis of 3D designs. Height attributes added to 2D building plans are used to produce 3D buildings. Symbols used in a 2D layout are linked to 3D model files which are automatically generated when users switch to 3D. Other components or product specials can be modelled using a sheet-based modelling approach.

MATERIALS HANDLING

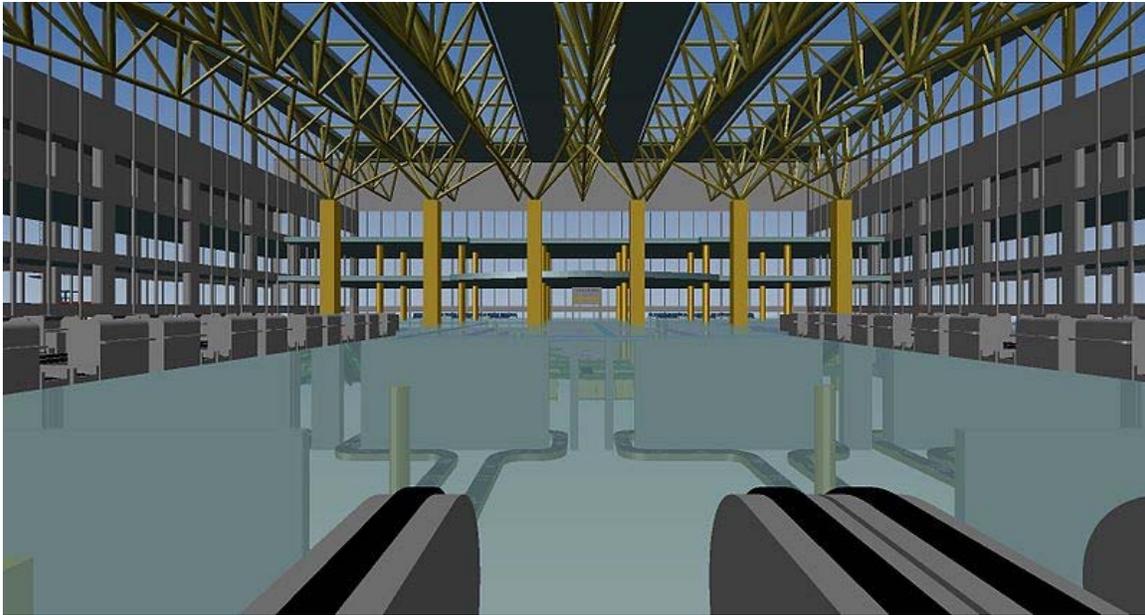
MPDS4 MECHANICAL HANDLING is a design application with a series of configurable catalogs of mechanical materials handling components, which can be physically interconnected to form part of an industrial process. It includes catalogs of conveyor belts, cranes, fork lift trucks, industrial racking and robots and allows installation designers to select, lay out, configure, visualise and add intelligence to process machinery in a plant. The module also supports the controlled creation of product specials for materials handling.

STEEL DESIGN

MPDS4 STEEL DESIGN is a module for constructing steel frames for buildings and equipment support. MPDS4 STEEL DESIGN includes catalogs of steel sections for many worldwide steel standards and allows users to design steel members, plates, stairs and ladders.

DUCTING DESIGN

MPDS4 DUCTING DESIGN is a software module for routing HVAC, of differing sections, into a plant or factory. MPDS4 DUCTING DESIGN includes catalogs with different types of ducts, valves, fans and other supporting components.



MPDS4 Engineering Review virtual walk-through of an airport

ELECTRICAL DESIGN

MPDS4 ELECTRICAL DESIGN is a design module for routing or connecting electrical systems with components throughout a plant, ship or factory. The user extensible and customisable catalogs contain many different types of electrical and control systems, as well as cable trays, cable ducts and links, and other supporting components. Auto routing functionality finds the shortest cable route between two designated points. By adding a KVA (kilo volt-ampere) rating to

selected components, users can analyse the required power rating of an entire network of connected instances and cables.

HANGERS & SUPPORTS DESIGN

MPDS4 HANGERS & SUPPORTS DESIGN is a design application for accurately modelling supports between pipes and steelwork in a plant or installation.

ENGINEERING REVIEW

MPDS4 ENGINEERING REVIEW is an application for conducting realistic engineering design reviews within the MPDS4 plant environment, visually presenting all of the plant project data. The module contains functionality for sectioning and setting transparency and allows users to define and generate movie-like walkthroughs of an installation.

REVIEW

MPDS4 REVIEW is an external review application for users who do not have the MPDS4 plant design system installed. MPDS4 can generate very compact .HSF (Hoops format) files of a plant design, which can be e-mailed to users of the MPDS4 REVIEW tool. They use the software to conduct interactive design reviews and walk-throughs, or present designs to third parties.