

Chapter 5

Civil Engineering Software Development at MIT

Author's note: I worked as an undergraduate and graduate research assistant in the MIT Civil Engineering Department from September 1957 to June 1961.

While MIT is often seen as a major center of research regarding mechanical engineering design software it was also where significant early civil engineering software was developed. The bulk of this work was done in the late 1950s through the mid-1960s under the leadership of Professor Charles L. Miller. He was one of the first to see the potential of the relatively new computer when he joined the faculty at MIT in 1955 as a 25-year old assistant professor of surveying. In this role he soon became head of the MIT Photogrammetry Laboratory. Surveying instruction at MIT began changing under his guidance from its traditional instrument orientation to teaching students how to process and analyze spatial data.¹

Surveying as a technical skill began evolving in the early 1700s at the same time that forerunners of modern instruments such as the transit and level became available. One of the first surveyors of repute in the United States was a young George Washington. The vast open spaces of the American continent led to the development of a rectangular grid system for the emerging United States and served to foster a growing surveying profession. To a great extent, surveying and civil engineering were closely intermingled until the early 1900s. As structural engineering, highway design and sanitary engineering



Figure 5.1
Professor Charles L. Miller and Students²

¹ Surveying had long been a key component of civil engineering education since many civil engineering graduates started their careers doing surveying work. Until 1950, MIT's Civil Engineering Department ran a surveying summer camp called "Camp Tech" for undergraduates in East Machias, Maine. To quote the department's Spring 2002 newsletter: "Installing a benchmark is no longer considered a mandatory job skill for graduates."

² MIT Civil and Environmental Engineering. Photograph is probably around 1965 or 1966.

became part of the civil engineering curriculum, surveying became more a specialty of its own. In recent years, aerial photography, laser distance measuring, data recorders and Geospatial Positioning Systems (GPS) have replaced traditional surveying instruments and the practice of professional surveying has become highly specialized.

New techniques for acquiring terrain data

When Miller first joined the MIT faculty there were two research areas that he felt needed to be explored. One was to develop better ways of acquiring spatial data with the focus being on utilizing new stereoscopy techniques and the other was utilizing emerging computer technology to process this data. Photogrammetry is basically the science of making spatial measurements using photographs while stereoscopy is the viewing of these photographic images in three dimensions. Using overlapping aerial photographs and a variety of projection devices it is possible to create contour maps and to measure three-dimension ground coordinates without having to physically survey the area except for establishing a small number of control points.

In the mid-1950s, the Photogrammetry Laboratory installed the stereoplotter shown in Figure 5.2. A pair of overlapping aerial photographic transparencies were placed in the unit's overhead projectors and carefully aligned so that a focused image was projected on the table. A viewing device enables the operator to determine elevations by adjusting a dot of light so that it appeared to be on the terrain surface. At the same time, a geared mechanism indicated the X and Y coordinates of that location. Recording a sufficient amount of data to be used as input for a computer program that calculated earthwork volumes was a very time consuming process. Under the direction of Dan Schurz, a graduate student, the laboratory began building a device that would convert stereoplotter data to three-dimensional coordinate values and output that data using a keypunch machine. This device became operational around 1959 or 1960.

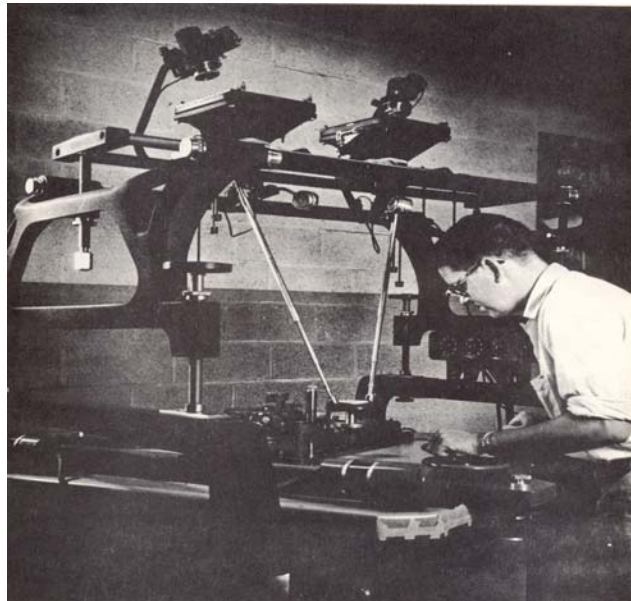


Figure 5.2

MIT Photogrammetry Laboratory Stereoplotter Being Operated by Bob LaFlamme³

The first software package developed at MIT about which I have been able to find information was a Borrow Pit Program (i.e. calculate the volume of material removed from a gravel pit or similar such area) written for the IBM 650 computer in 1956 by Paul O. Roberts⁴ who was a research assistant at the time. He worked with Vincent J. Roggeveen, an assistant professor of transportation engineering. It does not appear that Miller was engaged in the department's earliest computer activity. The IBM 650 was a drum computer that had less computational power than a current cell phone. Data input was in the form of 80-column punch cards and the output was also on punch cards. This computer was slow, awkward to use and dependent upon have the punch cards in the correct sequence, but it was a start.

Roberts' work was part of a Joint Highway Research Project between the MIT Department of Civil and Sanitary Engineering⁵ and Commonwealth of Massachusetts Department of Public Works. The Borrow Pit Program used data directly from surveying field books keypunched into punch cards. This data was read by the computer and the terrain profile for the original ground and the current level of the borrow pit were determined. The area of each cross section was calculated and the volume was calculated using a technique called the average end area method. The results were punched into cards by the 650 computer and printed on a machine that in those days was called a tabulator. The input punch cards contained 80 columns of data and a panel with control wires was needed so that the 650 computer could understand what the different fields of data represented. A similar type of control panel was needed for printing the program's output. In a few minutes, this program handled computations that would have taken a technician many hours to accomplish.⁶

Miller focused the Photogrammetry Laboratory's efforts initially on the development of a series of highway design programs built around the concept of a Digital Terrain Model and the building tools for acquiring digital terrain data either photogrammetrically or by digitizing existing contour maps. By 1956 the U. S. Bureau of Public Roads (the forerunner of today's Federal Highway Administration) was funding a significant portion of this work. One device built by Phil Gladding, with some help from me, was a device for recording terrain values from contour maps and punching this data directly on punch cards. See Figure 5.3.

Digital Terrain Model concept

The DTM concept was a significant breakthrough in how engineers thought about highway location and the work that went into establishing horizontal and vertical alignments and calculating earthwork quantities. The traditional manual approach for highway design involved selecting a preliminary horizontal alignment and the acquiring terrain elevations along cross sections perpendicular to this alignment from either field surveys or existing contour maps. This technique had evolved over nearly 100 years and

³ Miller, C. L. and LaFlamme, *The Digital Terrain Model – Theory and Application*, MIT Photogrammetry Laboratory Publication 117, March 1958

⁴ As an assistant professor of civil engineering, Roberts was the my masters thesis advisor in 1961

⁵ Now the Department of Civil and Environmental Engineering

⁶ MIT Joint Highway Research Project Research Report No. 20, Revised Second Edition, September 1957

worked fairly well as long as no significant adjustments were made to the alignment. If they were, then new cross section terrain data had to be acquired.



Figure 5.3
Digital Terrain Data Recorder⁷

The DTM method was predicated upon the concept that the surface of the ground could be statistically represented by a large number of XYZ data points. These could be based either on an existing coordinate systems such as a state plane system or an arbitrary coordinate system defined for a given project. Fundamentally, all contemporary highway design applications use this approach today. The implementation used at MIT during this period involved establishing a project-specific baseline and then recording terrain data along scan lines. The elevation data was recorded at either given intervals along the scan line or at predefined elevations corresponding to contour elevations on a hard copy map.

Miller and his team of research assistants understood the statistical significance of terrain data and the fact that small random errors would not affect overall results. The first series of highway design programs developed at MIT were called the Digital Terrain Model System. There were nearly a dozen separate programs in this suite of software including terrain data editing, horizontal alignment, vertical alignment and earthwork calculation. The reason for the large number of programs had to do with the fact that they were written for the IBM 650 computer which was the typical machine used by state highway departments at the time. The 650 had limited capacity to store programs and data. As a result, each program had to be loaded into the computer each time it was run along with appropriate design and terrain data. The punch card output of one program became to input to the next program.

As an example, one program was used to calculate the centerline of a highway using Points of Intersection (known as P.I.s or the coordinates of where two straight sections of the highway theoretically intersected) and the radii of the curves associated with those P.I.s. The output of this program was a complete definition of the roadway

⁷ *Technology Review*, June 1959

centerline. The output was then used by a different program to calculate the geometry that defined offsets from the centerline such as the edges of a median or the outer edges of the roadway. The next step was to use this data to calculate the vertical alignment of the roadway. Finally, earthwork volumes could be calculated by another program.

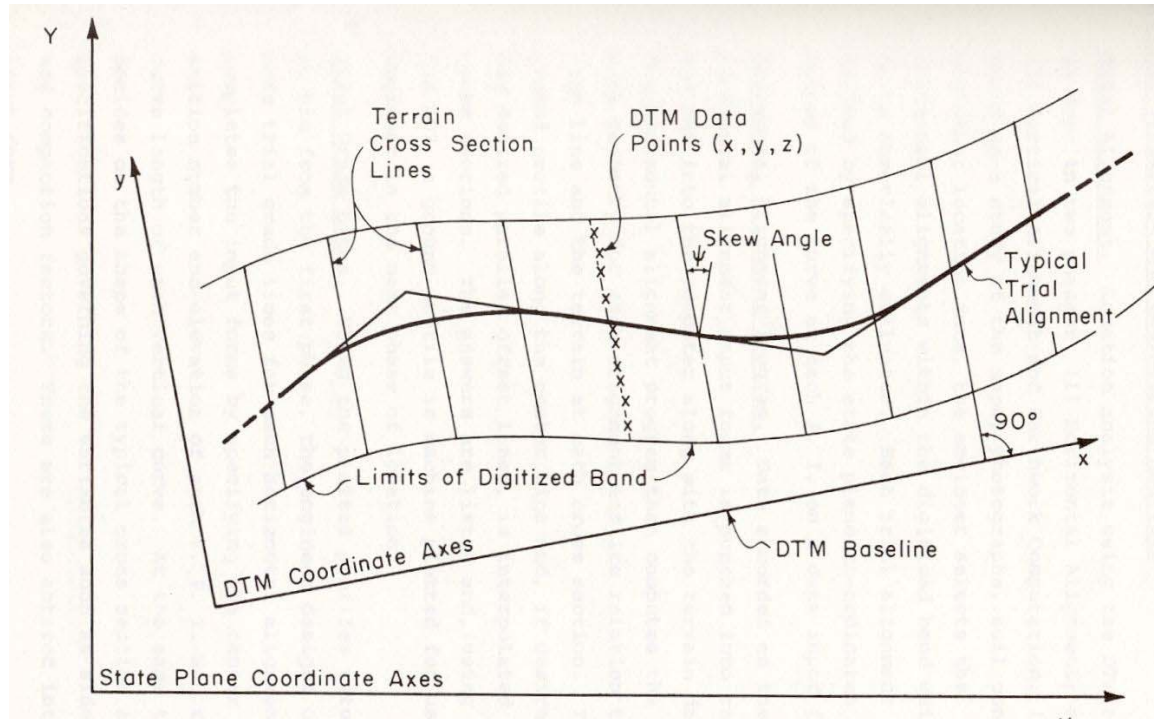


Figure 5.4
Digital Terrain Model⁸

These programs went through several iterations and by 1960 a new suite intended for highway location analysis and preliminary design was also available from MIT. Known as DTM II, it consisted of a Terrain Edit Program (TD-5), a Horizontal Alignment Program (HA-5) and a Roadway and Earthwork Program (EW-5). This software was designed by Paul Roberts and Bob LaFlamme with much of the actual coding done by Roger Baust, Dwight Rehberg and R. B. Doggett with testing done by an experienced highway engineer and surveyor, Ed Newman. Newman worked with Miller for over 30 years at MIT and CLM/Systems as discussed below.

The DTM System was far from a perfect solution - data acquisition was time consuming, the programs were slow and susceptible to errors in sequencing the input data and there were few machines available capable of plotting the output. All this began to change rapidly around 1960. The MIT Civil Engineering Department installed a higher speed computer, an IBM 1620, CalComp introduced the first plotter designed specifically for plotting digital computer output and new high level programming languages such as

⁸ *Digital Terrain Model System Manual of Electronic Computer Programs for Highway Location and Design*, November 30, 1960

FORTRAN became more readily available. Around 1960, the Photogrammetry Laboratory became the Civil Engineering Systems Laboratory.

Development of COGO

Miller is perhaps best known in the civil engineering community as the original developer of COGO – the coordinate geometry technology that is at the core of nearly all surveying and roadway design software today. COGO was one of the first examples of what is referred to as a “problem oriented” language. It enables a user to solve a wide variety of geometry problems by defining the interrelationships of points, lines, angles and curves. Interestingly, while Miller provided general direction for most projects under his supervision, COGO was his own personal project.

In simple terms, a COGO statement defined a specific geometric entity such as a point on the ground, a length between two points or an angle between two lines. A new point was calculated by telling the computer it was located a specific distance in a specific direction from a previously defined point. For example, a typical COGO statement might read:

LOCATE POINT 2 FROM POINT 4, DISTANCE 125.16, BEARING N45 15 20 E

Eventually, a shorthand version of COGO was developed so that this statement could be written:

LOC 2, 4, 125.16, N45 15 20 E

Far more complex series of calculations could be initiated including complete traverses and highway intersections. These statements were entered into the computer via punch cards or other means and the COGO program would sequentially step through the statements, calculate the requested data and save it.

An experimental predecessor to COGO was written for the IBM 650 and given the intriguing name of *Tricky Dicky Traverse*. I have never been able to determine where that name came from other than the fact that Richard Nixon was vice president at the time. Miller sketched out the basic concept for COGO on the back of an envelope one weekend and soon began implementing it on the new IBM 1620 computer.⁹ In the 1960 time period, Miller was also doing some consulting work for the Puerto Rico Bureau of Highways and the first version of COGO was installed in mid-1960 on a 20K character 1620 that they had recently installed. Subsequently, a research version was installed on MIT’s 1620. This was followed by implementations on Digital minicomputers and IBM mainframes. Copies of these various implementations were submitted to a number of software libraries and became public domain packages.

Interestingly, Miller claims that COGO was attacked by the “computer establishment.” He went on to state: “Only the users applauded. It was said that COGO would make it possible for *just anyone* to use the computer. In essence, the opposition to COGO was that it was too *user friendly*...”¹⁰

⁹ The MIT Civil Engineering Systems Laboratory received one of the first 1620 computers produced by IBM. It arrived damaged and was replaced a few weeks later by a new machine. Interestingly, the shipping container had NASA’s Marshall Space Flight Center marked out and the shipping label changed to MIT.

¹⁰ Miller, Charles L., *The COGO Story – An Odyssey*, CLM/Systems Publication, 1989

The concept of problem-oriented languages began to be explored by others. Independently of MIT, Dr. Steve Fenves implemented a structural engineering program, STRESS (STRuctural Engineering System Solver) that used a similar problem-oriented language methodology.

Development of ICES

Although he was just an associate professor without tenure and had never earned a doctorate, Miller was made chairman of the Civil Engineering Department in 1961, a position he held until 1969. At 32, he was the youngest person to ever hold this position and quickly set out to bring the department into the modern computer age. One of his first steps was to bring Fenves on as a visiting professor.

The department's early efforts to create a new highway design methodology eventually led to a major development project begun in 1964 called ICES or Integrated Civil Engineering System¹¹, which included popular programs such as ROADS, STRESS and STRUDL as well as COGO. The MIT development personnel were strong proponents of problem oriented languages and that focus continued with the ICES project. Led by Daniel Roos and Joe Sussman, the development team created its own programming language, ICETLAN, a civil engineering variant of FORTRAN and an engineering software-oriented operating system. Miller liked to refer to this group of undergraduates and graduate students as his "COGO kids."

A basic ICES premise was that in order for an engineer to address a complete problem solution, the results from one application task needed to be available as input to a subsequent task. Discussing an engineer's use of ICES Roos commented: "At any point in his problem solution he can leave one subsystem, enter another to perform calculations and then reenter the original subsystem using the results just obtained."¹²

Each application program (subsystem in MIT terminology) enabled an engineer to define a series of tasks that were to be applied to that problem's data set and to do so in terms that were meaningful to the engineer. Much like geometry problems could be defined with COGO statements such as what was shown above, structural, soil engineering or highway design problems could be defined using terms relevant to that type of engineering. One advantage of this approach was that if data items changed or the engineer wanted to change the problem definition, these source statements could be easily edited and the problem re-run.

An application subsystem consisted of a series of subroutines that executed the tasks defined by each problem statement. These were not huge monolithic programs. Rather they were a series of software modules written in ICETLAN. Furthermore, ICETLAN itself was not a software compiler but was what programmers call a "pre-compiler." To create an application subroutine, a programmer would write the necessary code in ICETLAN which would then be converted by another program into standard FORTRAN source code statements. That FORTRAN code was compiled to create the application subroutine. ICETLAN software was also implemented to handle the

¹¹ In addition to the U. S. Bureau of Public Roads and the Massachusetts Department of Public Works, ICES sponsors included IBM Corporation, the National Science Foundation, McDonnell Automation Company and the Ford Foundation.

¹² Roos, Daniel, *An Integrated Computer System for Engineering Problem Solving*, AFIPS Conference Proceedings, Volume 27, Part 2, Thompson Book Company, 1967 Pg. 152

management of complex data arrays, a task that the basic operating systems at the time did not do very well.

To solve a design problem, the engineer would define the data and the tasks to be applied to that data in a series of problem oriented statements. The ICES executive program processed these statements, checking for errors and inconsistencies. The software would then call individual subroutines to execute the statements. One statement was completely processed before the next one was executed, although this was typically transparent to the user.

By the mid-1960s, civil engineering software development at MIT was being done on an industrial strength computer, an IBM System 360 Model 40 with a 128K (32-bit words) memory and a pair of disk drives. Rather than using punch cards, programmers and application users began using alphanumeric terminals although early versions of ICES applications still envisioned the use of punch cards for data input and printed output.

STRUDL (STRUctural Design Language) development was led by Professor John Biggs¹³ and Robert Logcher who later became a professor at MIT. This software, which was first made available to users around 1967, was a significant extension to the earlier STRESS program in that it incorporated ICES capabilities for managing data and other functions. Users could define two and three dimensional framed structures with rigid or pinned joints. An engineer could define the basic structure, perform a preliminary analysis and subsequently refine the structural design by simply changing the location of joints or the size and orientation of members. A simple STRUDL statement might read:

JOINT 2 COORDINATES X 10.5 Y 20.6

As with COGO and other ICES programs, a shorthand version of these commands was also provided. Analysis output was stored so that the design engineer could request that additional information be printed without having to rerun the analysis.¹⁴

ICES ROADS was developed during the same timeframe as STRUDL. The key individual responsible for ROADS was John Suhrbier assisted by John Prokopy, Edward Sullivan and Wayne Pecknold. ROADS consisted of four major modules. The first handled the creation of a terrain database, a necessary step in order to do any design work. While there were specific commands for inputting terrain data, other programs could also format bulk input of this data. The second module was for defining the horizontal and vertical alignment of the proposed highway while the third module was used to define roadway cross sections and calculate earthwork volumes. These modules worked closely with ICES COGO and, in fact, many COGO commands were duplicated within ROADS although there was only a single copy of the COGO code loaded on the computer system being used.

The fourth module was for the simulation of vehicle performance. A number of commands were available to describe the subject highway including lane descriptions, traffic signals and intersections. The user could then define the types of vehicles and the

¹³ Professor Biggs was my structural engineer instructor the late 1950s.

¹⁴ Biggs, John M. and Logcher, Robert D. – *ICES STRUDL I – Structural Design Language – General Description* – MIT Structures Division and Civil Engineering Systems Laboratory, September 1967

traffic volumes at different times. The software would then calculate expected vehicle operating costs and average speeds. Overall, the command language for ROADS was far more complex than for most other ICES applications. One surprising aspect of ROADS was that there did not seem to be any way of producing plotted output from this package other than what were called “character plots” on an alphanumeric printer.¹⁵

Gradually more applications were added to the ICES system including TRANSET for transportation network analysis, SEPOL and LEASE for soils engineering, BRIDGE for bridge design and PROJECT for project management.

One of the strengths of ICES was that users could extend the capabilities of the individual programs using the same software development tools used by the system developers - not unlike the use of Bentley Systems’ MicroStation Development Language (MDL) years later. The major shortcoming of ICES was that it did not incorporate interactive graphics - it was a few years too early for that technology to be practical.¹⁶

Miller’s observation of the impact ICES is interesting:

“ICES achieved considerable success as an advance in applying computer technology....However, as an integration of the civil engineering profession, ICES was not very successful. Some observe – perhaps correctly – that I confused ICES with my attempts as department head to reorganize and revitalize the civil engineering department at MIT – a partial, through controversial, success.”¹⁷

Miller moves on from MIT

In 1968 Miller became head of MIT’s Urban Systems Laboratory and continued as director until 1977. Also in 1968, Miller was appointed by President Nixon to head a Transportation Task Force. By the late 1960s the ICES project started winding down as people such as Roos, Sussman and Logcher moved on to new challenges. Miller was named associate dean of engineering in 1970 and was the interim director of the Charles Stark Draper Laboratory for one year while it was undergoing the transition from being the MIT Instrumentation Lab to an independent research facility.

In 1977 Miller left MIT and focused his activities on CLM/Systems, a software and civil engineering consulting firm he had earlier established in Tampa, Florida. The company developed a series of civil engineering software applications including COGO, TOPO for processing topographic data and ROADS for highway design. In 1986 these were integrated together in a product called CLM CEAL (Civil Engineering Automation Library) which was used by a number of state highway departments and civil engineering firms.

CLM/Systems actually started out as CLM/Research in 1955 when Miller first joined the MIT faculty and needed a vehicle for doing consulting work beyond the research activities of the Photogrammetry Laboratory. The name was changed to

¹⁵ Suhrbier, John H., et al – *ICES ROADS I – Roadway Analysis and Design System – Engineer’s Reference Manual* – MIT Civil Engineering Systems Laboratory, March 1968

¹⁶ Roos, Daniel, *An Integrated Computer System for Engineering Problem Solving*, AFIPS Conference Proceedings, Volume 27, Part 2, Thompson Book Company, 1967 Pg. 151

¹⁷ Miller, Charles L., *The COGO Story – An Odyssey*, CLM/Systems Publication, 1989

CLM/Systems in 1968. Until about 1981 the company was primarily a consulting company working on urban planning studies and assisting clients with implementing computer technology. At that point, Miller decided that there was a need for a new generation of civil engineering software and set out to create it.

The company soon grew to about 30 people, none of whom were classified as sales or marketing. The software was sold primarily by word-of-mouth. CEAL enthusiasts existed in numerous state and county highway departments including Georgia, Washington, New York, Dallas and Los Angeles. For a period of time, McAuto sold CEAL as an alternative to the MOSS software from MOSS Systems in England which it had supported since the mid 1980s. CLM/Systems also had a close working relationship with Intergraph and even sold a product called CEALstation which was a combination of CEAL and MicroStation.

For more than two decades after leaving MIT, Miller was still referred to as “Professor.” He always wanted to be the teacher. Following are his guidelines for the successful use of CEAL – they could apply to any engineering automation software package.

- “Don't try to force CEAL into being something it is not and don't wait for the next release. USE CEAL AS IT IS.
- If you try to make CEAL act like some other package you are acquainted with, you will fail. Every software package, like every person, has its own 'personality.' You cannot change it without grave consequences.
- Don't fight the system. Learn to use it on its own terms. You can be creative, clever, imaginative, and capable with CEAL, but you cannot be arrogant.
- CEAL is a very dynamic system under continuous development. There will always be another release on the way. But, to wait for it would be fruitless. You will wait forever for the next release.
- In the meantime, the current release of CEAL contains more capability than any user can master in a lifetime. Best to get on with using the tools at hand.”¹⁸

By the mid-1990s CEAL had become a well rounded civil engineering application that was available on a wide range of engineering workstations and DOS-based PCs. By 1995 the software sported a graphics interface although it was not up to speed with competitive products in this regard. The price for CEAL (including COGO) ranged from \$8,000 for a single license to just \$1,500 per license on orders for more than 100 copies.

While there never was any question about the quality of this software, the company rarely had revenues much over \$1 million in the early 1990s (earlier it had revenues over \$2 million when it was selling turnkey systems consisting of both computer hardware and software) and by 1996 it had slipped into a death spiral. There

¹⁸ *A-E-C Automation Newsletter*, December 1990, Pg. 5

was an attempt to sell the company as Miller's health was starting to fail. Unfortunately, there were no takers and the company just withered away over the next several years.¹⁹

There were probably several hundred different versions of COGO implemented by a vast array of companies and organizations, most of whom used the original version as the starting point for their development. None of these developers paid any royalties to Miller who passed away in 2000.

Many of the early participants in the DTM and ICES activity including Trond Kalstaad, Robert Logcher, Dan Roos and Joseph Sussman, stayed at MIT throughout their careers and contributed significantly to the Institute's academic excellence. Others left and expanded Miller's ideas throughout the engineering profession. Leroy Emkin went to Georgia Institute of Technology where he expanded the capabilities of ICES STRUDL and turned it into the very successful GTSTRUDL program. Barry Flachsbart went to work as manager of analysis and development at McDonnell Automation (McAuto) which licensed the ICES software and sold it in a timesharing mode well into the 1980s.

As described in Chapter 19, McAuto was probably the largest seller of ICES services. The company added dynamic capabilities to STRUDL with a program called STRUDL-DYNAL which was used to design numerous structures including the Louisiana Superdome in New Orleans, off-shore oil and gas platforms and nuclear power plants. The company implemented enhancements to ROADS and COGO and added a sanitary and storm sewer design program to the ICES suite simply called SEWER. By 1975, many McAuto customers were using graphics terminals such as the Tektronix 4010 and 4014 to interact with these ICES programs in a time-sharing mode.

McAuto also implemented a graphics program call FASTDRAW that enabled users to create input data for programs such as STRUDL and view plots of the results. Time-sharing use of STRUDL could end up being quite expensive. A complete STRUDL-DYNAL analysis of a large structure (800 joints and 950 steel members) could cost as much as \$4,000.²⁰

One of the most significant aspects of Miller's work and why I feel he deserves greater recognition than he has received is that he never considered the MIT version COGO to be proprietary technology – he made it readily available to the world without any restrictions. The only other similar example I can think off is Tim Berners-Lee the creator of the World Wide Web. Imagine where we might be today if these two pioneers had decided to patent their technology and required us to pay a royalty every time we designed a highway intersection or used the Web.

I was very please to accept the 2002 Ed Forrest Award on behalf of Professor Miller and his family at the A/E/C SYSTEMS 2002 conference in Dallas, Texas. This award, named after the founder of *A-E-C Automation Newsletter*, was awarded annually to an individual(s) who had made a significant contribution to the field of AEC software.

¹⁹ I was hired by Miller to find a buyer for his company but was unable to do so.

²⁰ Dallaire, Gene, *The CRT Computer Graphics Terminal: Indispensable Design-aid for Some Structural Engineers*, Civil Engineering, February 1976