MPC REPORT NO. 92-9

ENHANCEMENT OF EXISTING ENGINEERING SOFTWARE

VOLUME NO. 1

Linking Computer Aided Engineering Procedures with Computer Drafting

J. A. Puckett
Chad Clancy

Department of Civil Engineering
University of Wyoming
Laramie, Wyoming 82071

July 1992

ACKNOWLEDGEMENT

This report has been prepared with funds provided by the United States Department of Transportation to the Mountain-Plains Consortium (MPC). The MPC member universities include North Dakota State University, Colorado State University, University of Wyoming and Utah State University.

PREFACE

The two major objectives for this report were to develop effective methods for:

- Enhancing the user interfaces for existing transportation software.
- Linking design applications directly to computer-aided design.

This document outlines the results obtained in meeting the second objective. This document is written for design professionals who may not be familiar with programming procedures. The authors' intent is to provide the necessary information for the reader to assess the usefulness of the UWGRAPH library and determine whether the library is applicable for his CAD development work. Complete technical documentation is in Volume 2 *"UWGRAPH Computer-Aided Design and Drafting Library."*

We believe that UWGRAPH will help other software development professionals. We hope that applications developed with UWGRAPH applications can be shared throughout the transportation design industry. Further, carefully documented utilities can be used by all developers. We encourage UWGRAPH users to develop documented utility procedures and to share these procedures with others.

ABSTRACT

Often in traditional design procedures, a designer sketches results which are based on computer applications and transmits this information via hardcopy to a drafter, who reenters it into a CAD system. The inefficiencies are clear and an obvious opportunity exists for productive gain. Recent work by the Wyoming Department of Transportation (WDT) has shown that productivity ratios greater than 20:1 can be achieved by linking design applications directly with CAD. A graphics library (UWGRAPH) has been developed to facilitate this linkage.

UWGRAPH is used in conjunction with a design application which performs engineering calculations and/or creates drawing parameters. UWGRAPH links to three graphic formats commonly used in engineering DXF (AutoCAD), MICRO-CSL (Intergraph) and GKS (screen graphics) in one application program. The flexibility and familiarity of FORTRAN is combined with the tools necessary to produce graphics files automatically. The library is based on the graphics entities required for structural drafting. UWGRAPH simplifies and unifies the subroutine calls by combining low-level calls to produce graphical entities useful in engineering drawing, e.g., dimensions, notes, etc. The UWGRAPH scope is directed toward bridge engineering but easily spans other disciplines.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Often in traditional design procedures, a designer sketches results which are based on computer applications and transmits this information via hardcopy to a drafter, who reenters it into a CAD system. The inefficiencies are clear and an obvious opportunity exists for productive gain. Recent work by the Wyoming Department of Transportation (WDT) has shown that productivity ratios can exceed 20:1 by linking design applications directly with CAD. A graphics library (UWGRAPH) has been developed to facilitate this linkage.

UWGRAPH is used in conjunction with a design application which performs engineering calculations and/or creates drawing parameters. The flexibility and familiarity of FORTRAN is combined with the tools necessary to produce graphics files either to the computer screen or to a drafting/design file which can be used by a CAD system. The library is based on the graphical entities required for structural drafting.

Many existing graphic systems are difficult to learn and implement. UWGRAPH simplifies and unifies the subroutine calls by using low-level calls combined with intermediate calculation to produce graphical entities useful in engineering drawing, e.g. dimensions, notes, etc. Thus, only one call is used in UWGRAPH for each entity. All the calculations and decisions involved with leader type and placement, text justification and location, etc. are automatically performed in the desired graphical format. Hence, UWGRAPH performs a great deal of work for the program developer.

Standardization Efforts and Objectives

There has been much effort to standardize computer graphics. CORE, GKS, PHIGS, DXF are examples of standardized graphics library and file formats (6). But creating an all-encompassing graphics format that suits every application, type of drawing, and computer system is difficult. Further, a single system which supports the development environment adequately may not support the production environment. For example, a library is used which produces a DXF file which must be interpreted by a CAD system prior to viewing the results. Because the file is converted to the native CAD format, this approach is well suited for production. However, it is an extremely cumbersome approach for the development engineer.

A library based solely on a screen/plot presentation system, like GKS or PHIGS, produces fine screen graphics almost instantaneously upon execution. This approach is amendable to both development and production engineers who wish to iterate design programs rapidly. However these libraries produce drawing files in formats required by only screen and plotting devices. These files cannot be easily edited by a drafter for inclusion in a set of plans. This is a major limitation of this programming approach as the greatest productivity potential lies with linking the application directly to CAD.

The intent of the UWGRAPH library is to merge the favorable attributes of the screen/presentation libraries with libraries which automate the linkage to CAD. Specifically, the objectives of the UWGRAPH development effort is to

develop a graphic library to:

- ✓ Produce structural drafting which can be extended to other disciplines.
- ✓ provide an environment for rapid program development.
- ✓ Provide an environment fast enough for the design engineer iterating on design solutions.
- ✓ Produce a file which can be interpreted or used directly in CAD systems.
- ✓ Use existing standardized graphics and CAD formats linked to a common library.

LIBRARY ARCHITECTURE

The UWGRAPH library links to three graphic libraries commonly used in engineering. The purpose of the library is to allow the use of DXF, MICRO-CSL and GKS libraries through one application program. The library architecture is illustrated in Figure 1. Graphics can be created in one or more of these three formats from a single set of subroutine calls. This is helpful because the graphical output of the application program can be used by anyone who can display graphics in any or all of the systems. A simple software switch can direct the application to the desired format. The application programmer doesn't have to learn more than one graphics system because the initialization, drawing definition, and termination is defined consistently by the UWGRAPH library.

Further, the application developer can use GKS to review drawings in development without accessing the CAD system. This permits program development on inexpensive graphics devises and can help to eliminate scheduling conflicts on the CAD systems.

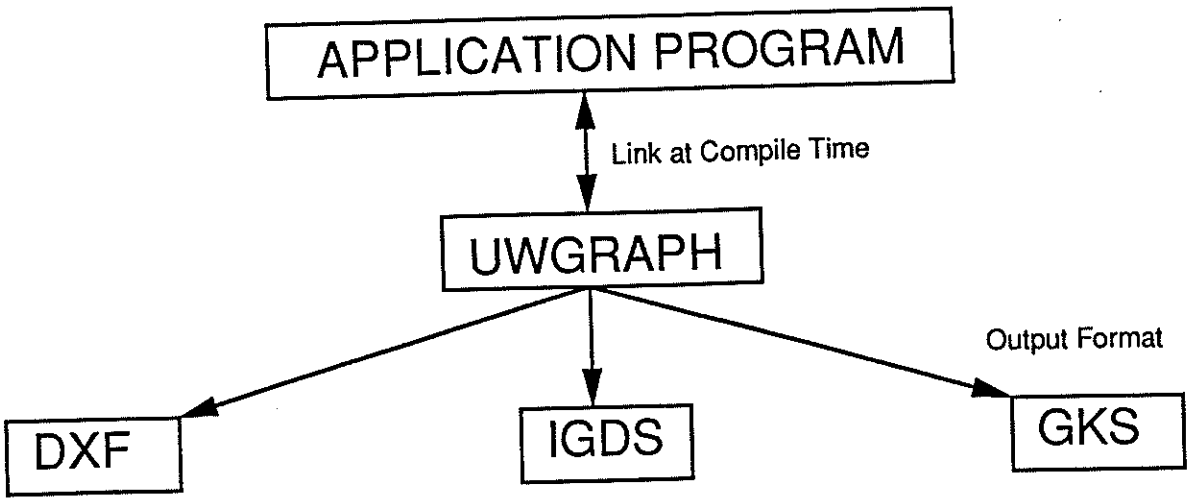

Figure 1. UWGRAPH Architecture

TABLE 1. PROCEDURES IN UWGRAPH (1)
Initialization: Initializes the appropriate graphics libraries for subsequent entity calls.
Line: Creates a two-point line.
Circular Arc: Creates an arc given a center, radius, start angle and an end angle.
Multiple Point Line: Draws a line string given more than two points.
General Note: Places strings of text.
General Dimension Call: Creates a dimension with a break in the leader to place the dimension text (if there is enough room for the text between witness lines.)
Second Dimension Type: Creates a dimension which has a leader arrow drawn from witness line to witness line with no break and a string of text or a numeric dimension placed at user defined coordinates.
Subset Version of UWDIM2: Creates a dimension similar to UWDIM2, but allows more than one text string to be placed.
Label: Creates a text label with two-segment leader with an arrowhead at the tip and up to three text strings at the leader end. Text justification is controlled in several ways.
Subset Version of Label: Creates same label as above but automatically draws a bracket around the text associated with the label.
Termination: Terminates all drawing processes.

TABLE 2. HIGH LEVEL UTILITIES (1)
Drawnut: Uses lines and arcs to draw an elevation of a nut of width NW and of height NH. The nut can be drawn with the flat side down or the flat side up.
Drawthr: Uses lines to draw bolt threads of width TW and of height TH. The threads can be drawn with a taper at the top, a taper at the bottom or with no taper. This utility will draw the threads with a vertical axis only.
Boxcoord: Fills the X and Y arrays with the coordinates to draw a rectangle.
BLDDIM: Builds two- or three-part dimensions separated by 'x' in a fractional format. For example: 1' 6 3/4" x 10 5/16" x 2".

EXAMPLES

Two examples are presented. The first is a simple example which illustrates the usage of UWGRAPH. The example, called testbox, performs most of the operations required in a typical drawing. The FORTRAN listing of testbox is given in Appendix A and the output is illustrated in Figure 2. The drawing process has three distinct parts: initialization, drawing, and termination. The initialization is performed with a single call of UWINIT, which initializes the target system. The termination is also performed with a call to UWTERM. In between initialization and termination, any drawing entity may be called. The testbox program uses line, text, and dimension entities.

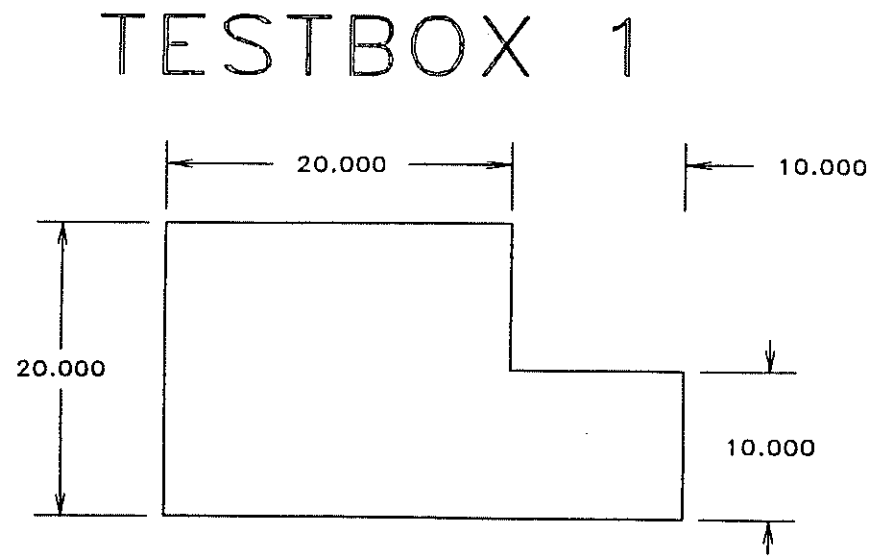

Figure 2. Output From Program Testbox

The second example is the bearing detail shown in Figure 3. This design detail was selected for automation because of its limited scope and the existence of an application to perform the design. This was the initial effort to test UWGRAPH in a production environment. In this application, a few design parameters are entered to a design program which generates the details necessary for drafting. The library is used to automatically generate the drafting file. This illustration was developed from an MICRO-CSL (Intergraph CAD) file. The generation of the design and drawing took approximately five minutes plus ten minutes for cleanup and merging with the project plans. If the design had been performed with the same application but the drawing "manually" illustrated in the CAD system, the time required is approximately 4.5 hours, which gives a productivity ratio of approximately 18:1.

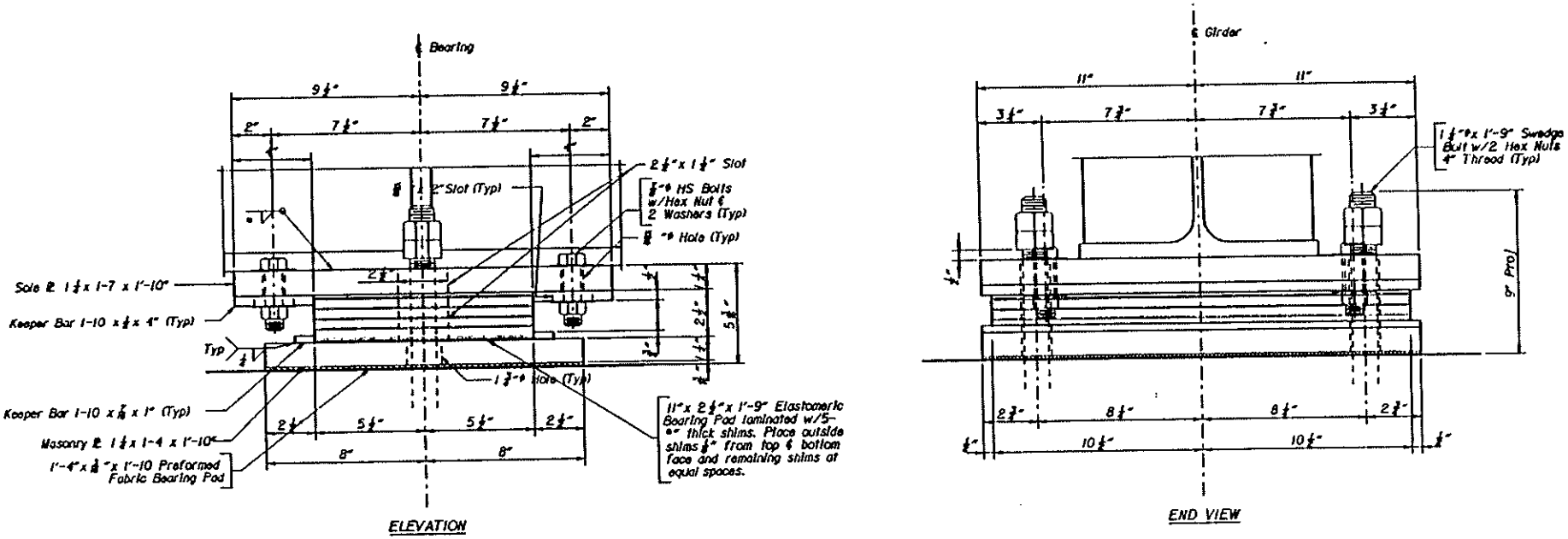

Figure 3. Bearing Details

CONTINUING WORK

UWGRAPH is presently being used to create design and graphical illustrations for reinforced box culverts, bridge geometry, screed elevations tables and web cutting diagrams. The initial work on the box culvert application was performed using the IGES format (3) and is implemented in the BRASS system (2). The bridge geometry program is presently under development. Code writing is underway at the time of this writing. This program includes general plan/elevation drawings and substructure layout. The screed elevation and web cutting diagram program is completed and is being tested by WDT engineers.

ACKNOWLEDGEMENT AND AVAILABILITY

The UWGRAPH library was developed under the sponsorship of the Wyoming Department of Transportation and the Mountain-Plains Consortium at North Dakota State University which is sponsored by the United States Department of Transportation. The items presented in this paper do not necessarily reflect the viewpoint of the project sponsors.

The UWGRAPH source code is available upon request. The code has been successfully ported to two versions of Unix, Integraph CLIX3.1 and Apollo Domain SR10.3. Other ports are being investigated.

All applications developed as part of this project are also available upon request. These include BRASS-Culvert, BRASS-Geometry, Screed, and BRASS-Bearing. Other example applications appropriate for development examples are included in the UWGRAPH documentation (1).

REFERENCES

Clancy, C. and Puckett, J.A., *UWGRAPH Computer-Aided Design and Drafting Library,* Mountain-Plains Consortium, North Dakota State University, Fargo, North Dakota, July 1992.

Puckett, J.A. and Guenther, P.W., Linking Civil Engineering Design and Drafting Software via IGES, *Journal of Computing in Civil Engineering,* ASCE, Vol. 3, No. 3, July 1989, pp. 228.

National Bureau of Standards, *Initial Graphics Exchange Specification,* Publication PB86-199759, Version 3.0, 1986.

Intergraph Corp., *Intergraph GKS/C Reference Manual,* DSYS027, Third Edition, Huntsville, Alabama, August 1988.

Hopgood, F.R.A., Duce, D.A., Gallop, J.R., and Sutcliffe, D.C., *Introduction to the Graphical Kernal System (GKS),* Second Edition, Academic Press, Harcourt Brace Jovanovich, London, New York, 1986.

Dewey, B.R., *Computer Graphics for Engineers,* Harper and Row, New York, NY, 1988.

APPENDIX A

```
      PROGRAM TESTBOX
C
C     This program draws a box and dimensions it using the
C     UWGRAPH library of graphics calls.
C
C     First, declare the working variables with this include statement.
C
CMSF$INCLUDE:'VARI.FOR'
CVAX       INCLUDE 'VARI.FOR'
CUNX       INCLUDE 'VARI.FOR'
C
C     Now, open up a file to output messages.
C
      OUTNUM=5
      OPEN(UNIT=OUTNUM,FILE='MESSAGE.OUT',STATUS='UNKNOWN')
C
C     Set up the initialization parameters.
C
      OUTNAME = 'BOXOUT'
      XWMIN   = 0
      XWMAX   = 90
      YWMIN   = 0
      YWMAX   = 90
      WKSID   = 1
      WSCON   = 1
      WSTYP   = 12
      UNIT    = 'FT'
CDXF       SYS      = 1
CITG       SYS      = 2
CGKS       SYS      = 3
      CALL UWINIT(OUTNAME,XWMIN,XWMAX,YWMIN,YWMAX,WKSID,
     1     WSCON,WSTYP,UNIT,IDBG0,SYS)
C
C     Initialize the default flags with this call.
C
      CALL USEDEF(ITYP1,ITYP2,ITYP3,ITYP4,ITYP5,ITYP6,ITYP7,
     1   ILEV1,ILEV2,ILEV3,ILEV4,ILEV5,ILEV6,ILEV7,
     2   ICOL1,ICOL2,ICOL3,ICOL4,ICOL5,ICOL6,ICOL7,
     3   WID1,WID2,WID3,WID4,WID5,WID6,WID7,
     4   SL4,SL5,SL6,SL7,
     5   ROT4,ROT5,ROT6,ROT7,
     6   SW4,SW5,SW6,SW7,
     7   HT4,HT5,HT6,HT7)
C
C     Now, some graphics calls can be made.
C
C     Draw a multiple point line.
C
      POINTS=7
      X(1)=20
      Y(1)=40
      X(2)=50
      Y(2)=40
      X(3)=50
      Y(3)=50
      X(4)=40
      Y(4)=50
      X(5)=40
      Y(5)=60
      X(6)=20
      Y(6)=60
```

```
      X(7)=X(1)
      Y(7)=Y(1)
      CALL UWMPLINE(X,Y,POINTS,ITYP3,WID3,ILEV3,ICOL3,IDBG3,SYS)
C
C     Place dimension at left side.
C
      DNUMX=14
      DNUMY=50
      XPT1=20
      YPT1=40
      XPT2=20
      YPT2=60
      HT5(1)=1
      CALL UWDIM(DNUMX,DNUMY,XPT1,YPT1,XPT2,YPT2,ITYP5,WID5,
     1      ILEV5,ICOL5,SL5,ROT5,HT5,SW5,IDBG5,SYS)
C
C     Place dimension at top-left.
C
      DNUMX=30
      DNUMY=64
      XPT1=20
      YPT1=60
      XPT2=40
      YPT2=60
      CALL UWDIM(DNUMX,DNUMY,XPT1,YPT1,XPT2,YPT2,ITYP5,WID5,
     1      ILEV5,ICOL5,SL5,ROT5,HT5,SW5,IDBG5,SYS)
C
C     Place dimension at top-right.
C
      DNUMX=45
      DNUMY=64
      XPT1=50
      YPT1=60
      XPT2=40
      YPT2=60
      CALL UWDIM(DNUMX,DNUMY,XPT1,YPT1,XPT2,YPT2,ITYP5,WID5,
     1      ILEV5,ICOL5,SL5,ROT5,HT5,SW5,IDBG5,SYS)
C
C     Place dimension on the right side.
C
      DNUMX=55
      DNUMY=45
      XPT1=50
      YPT1=50
      XPT2=50
      YPT2=40
      CALL UWDIM(DNUMX,DNUMY,XPT1,YPT1,XPT2,YPT2,ITYP5,WID5,
     1      ILEV5,ICOL5,SL5,ROT5,HT5,SW5,IDBG5,SYS)
C
C    Place a note.
C
      TCENX(1)=32
      TCENY(1)=72
      TOTCHR(1)=9
      NOSTR=1
      TBUFFR='TESTBOX 1'
      HT4(1)=4
      CALL UWNOTE(TCENX,TCENY,TOTCHR,NOSTR,TBUFFR,ITYP4,WID4,ILEV4,
     1      ICOL4,SL4,ROT4,HT4,SW4,IDBG4,SYS)
C
C     Now, terminate graphics creation.
C        CALL UWTERM(IDBG8,SYS)
        STOP
        END
```