Memorandum 28

CAMA: DEFINE PROBLEM COMMAND

Suzanne D. Goodrich

CONCOMP: Research in Conversational Use of Computers
F.H. Westervelt, Project Director
ORA Project 07449

supported by:
ADVANCED RESEARCH PROJECTS AGENCY
DEPARTMENT OF DEFENSE
WASHINGTON, D.C.

CONTRACT NO. DA-49-083 OSA-3050
ARPA ORDER NO. 716

administered through:
OFFICE OF RESEARCH ADMINISTRATION ANN ARBOR

June 1970
Enqk
UMKL
1675
# TABLE OF CONTENTS

1. Introduction .................................. 1
   2.1. Data Structure .......................... 2
       2.1.1. Master Directory Entries ......... 2
       2.1.2. Data Packs ......................... 4
   2.2. General Execution Structure ............ 5
   2.3. Subroutines Used in DPROB Command ... 22
3. References ................................. 31
1. INTRODUCTION

The Computer-Aided Mathematical Analysis system (CAMA) permits construction of symbols and expressions, manipulation of expressions, and solution of equations. CAMA operates under MTS in the IBM 360/67 and RAMP in the PDP-8/338 terminal. CAMA operations are divided into tasks which may be inserted into the queue via commands.

2. USER'S GUIDE FOR COMMAND DPRØB

The command DPRØB is used to define a problem. Execution of this command causes a flag to be set so that a second specification of the command is ignored until the first request is completed. The command has four branches: (1) construct symbols, (2) select symbols for a problem, (3) define expressions for a problem, (4) escape to return to the CAMA system.

(1) Construct symbols.

The user specifies the menu into which the generated symbol will be placed. He constructs and names the symbol. Options exist during construction for erasing the existing symbol and starting again or rejecting the entire stage. If the user duplicates a symbol name, he may rename the symbol or destroy the old definition as he wishes.

(2) Select Symbols for Problem.

The user specified a problem. If the problem
exists and contains symbols, the symbols are displayed. The user selects entire menus or particular symbols from menus to be included in his problem. He may remove one or all symbols from his problem. A maximum of 63 symbols may be included in one problem.

(3) Define Expressions for a Problem.
The user specifies a problem and the problem symbols are displayed. He may display, modify, and save or destroy an existing expression of the problem, or create and save a new expression.

2.1 DATA STRUCTURE
2.1.1. Master Directory Entries

The command DPROBJ uses entries MSYMTBL, MPROB, NAMEFLAG, and PROBLEM (see Fig. 1) in the master directory and the corresponding packs. MSYMTBL is a name-list containing the names of all menus, both system menus and user-named menus containing user-created symbols, and pointers to the menu packs. The menu packs are lists of symbols in the menu and pointers to the data pack which contains the display file for the symbol.

MPROBJ is a list of problems and corresponding pack pointers. A problem pack is an association table identifying the symbols in a problem. The association object is the symbol name, and the value is the name of the menu
Entries in
Master Directory

**MSYMTBL** → **MSYMTBL (name list)**
- **ALPHABET** ptr → **ALPHABET** (name list)
  - A ptr to DF data pack
- **NUMERALS** ptr
- **MENU1** ptr → **MENU 1** (name list)
  - A ptr to DF data pack
  - B ptr to DF data pack
- **MENU A3** ptr (Constructed symbols)
  - SYM1 ptr to DF data pack
  - THETA ptr to DF data pack

**MPRØB** → **MPRØB (name list)**
- **PRØBL** ptr
- **PRØB K** ptr → **PRØB K** (association table)
  - @ SYM (symbol name) (menu)
- **PRØB97** ptr
- **FLUID** ptr @ SYM A ALPHABET
- **TEST** ptr @ SYM THETA MENU1
- ...:

**NAMEFLAG** → **NAMEFLAG (association table)**
- **DPRØB** FLAG ØN
- **DPRØB** MENU (name)
- **DPRØB** PRØB (name)

**PRØBLEM** → **PRØBLEM (name list)**
- **PRØBL** ptr
- **PRØB97** ptr
- **PRØB K** ptr → **PRØB K (name list)**
  - EXPRESS1 ptr to expression data pack
  - FORM 3 ptr to expression data pack

Figure 1. Entries in Master Directory
containing the symbol.

The NAMEFLAG entry in the master directory points to an association table which identifies the current problem and menu, and indicates the availability of this command.

PROBLEM is a name-list of problems and the corresponding pointers. The problem packs are name-lists of expressions in the specific problem and pointers to the data packs defining the expressions.

2.1.2. Data Packs

The first ten words of the symbol display file data pack are the display file header (PSECT2 of the DF routines\(^1\)). Byte 1 of word 1 and words 6, the buffer location pointer, 7, the PDP-8 word count, and 9, the first location of the actual buffer, must be set to appropriate values before the display file is transmitted. Switch 1 which is byte 1 of word 1 should be set to 40\(_{16}\) so the DF routines will not modify the last location of the display file. The remainder of the data pack contains the display file with one 338 word stored in two bytes using six bits in each byte, as described and illustrated in Appendix C of The DF Routines User's Guide\(^1\).

Data packs are also used to define the expression with one 20-byte entry for each symbol of the expression (see Fig. 2).
The entry contains the X and Y coordinates of the symbol in raster points, the symbol name, the scale for the symbol in range 1-3, and the entity number. (The entity number is not used in command DPRØB and will be discussed in greater detail when relevant.)

2.2 GENERAL EXECUTION STRUCTURE

The general sequence of execution of command DPRØB is illustrated in Fig. 3. Successive tasks are executed in the central computer in response to light button requests until the user chooses to exit from the command.

Figure 3.
Task TKDPB

Task TKDPB examines the NAMEFLAG association table. If the association

DPRØB    FLAG    ØN

is present, a comment

TASK IN CURRENT USE ... NEW REQUEST IGNORED.

is printed and a return is made to the CAMA system. If the flag association is not present, the task DMAIN is executed.

Task DMAIN

The DMAIN task is the main branching stage. The light buttons

ESCAPE
PRØBLEM DEFINITION
CONSTRUCT SYMBØLS
SELECT SYMBØLS FØR PRØBLEM

are displayed. The user indicates the desired option with the Grafacon. If the user chooses to exit command DPRØB via the ESCAPE button, all DPRØB entries in the NAMEFLAG table are erased and a return is made to the CAMA system.

Task DPDEF

Task DPDEF is the response to the PROBLEM DEFINITION light button of task DMAIN. The execution of DPDEF is outlined in Fig. 4. The display screen is illustrated in
Fig. 5.

Task DCSYM

Task DCSYM is the response to the CONSTRUCT SYMBOLS light button. The 360 program which generates and transmits the display files is outlined in Fig. 6. The PDP-8 programs, specifically the actual symbol construction program, will be discussed in a subsequent memo. The display screen is illustrated in Fig. 7.

Task SSGTSSK

The SSGTSSK task moves the display file for the constructed symbol from a line directory to a data pack as described in Section 2.1.2. Execution of the task will be discussed in another memo.

Task SYSEL

Task SYSEL is executed as the response to the SELECT SYMBOLS FOR PROBLEM light button. The subroutine is outlined in Fig. 8. A maximum of 63 symbols may be selected for a problem. The display screen is illustrated in Fig. 9.
Find and confirm current problem name.

Check for problem name in MPROB list.
   name present
   name not present

Enter problem name in NAMEFLAG table.
Create temporary data pack to save expression which is formed.
Place problem name in negative region of temporary pack (required by display subroutine SE8).
Enter problem name in PROBLEM list.

Ready display, blank screen.
Transmit problem symbols as invisible subroutine files (subroutine INIT8).

Display symbols, 21/line.
Draw line above symbols to indicate expression display area.

Display light buttons.

Draw 3 guidelines in expression display area.
Reserve DF numbers.

Request and confirm expression name. User may request a list of all expressions in the current problem.

If the requested expression exists in this problem, copy the expression pack into the temporary data pack and display the expression.

Request a Grafacon hit on a light button.
Locate and mark the light button hit. Process the light button.

Figure 4. Subroutine DPDEF
LIGHT BUTTONS

PLACE SYMBOL WITH HALF/EQUAL/DUPLICATE SIZE  → Set switch=0  → 5
DESTROY → Print comment requesting confirmation of permanent removal of expression from data structure. → no (do not remove)  → Blank expression and light button mark. Empty temporary expression data pack.

ESCAPE → Blank screen. Free temporary expression pack. → Task DMAIN
REPLACE SYMBOL → Set switch=0  → 5
MOVE SYMBOL UP/DOWN/LEFT/RIGHT → Set switch=1  → 5
ERASE → Blank expression and light button mark. Empty temporary expression data pack.

DELETE SYMBOL AND CLOSE GAP/DELETE GAP → Set switch=1  → 5
INSERT/INCREASE/DECREASE GAP → Set switch=1  → 5

If a form of the requested expression exists in the data structure, the user is asked to confirm erasing the old form or to rename the current expression displayed.

SAVE → Enter the expression name into the problem list and create a data pack for the expression. → Reorder the expression as defined in the temporary data pack into the permanent data pack with increasing X then increasing Y (subroutine EXORD). → Blank light button mark.

MOVE WINDOW UP/DOWN/LEFT/RIGHT → Y

DISPLAY expression with new coordinates.  → Blank light button mark.

MOVE/REMOVE GUIDELINES → MOVE → Set switch=1  → 5
REMOVE → Blank guidelines and light button mark.  → Y

Figure 4. Subroutine DPDEF, continued
Wait for Grafacon hit.

- Hit above light button:
  - hit on light button → Blank current symbol mark → Y
  - Determine if hit on list of symbols or in expression display area:
    - Symbol hit:
      - Determine and mark symbol hit.
      - Is switch < 0? yes → ε
      - no → Set switch = 1
        - δ
      - ε
    - Expression display area hit:
      - Is switch > 0? yes → ε
      - no → Mark position hit.
        - Set switch = -1.
          - δ
  - Determine symbol hit.
    - Process the light button.

Note: Switch is used to indicate a symbol hit → set to +1, a display position hit → set to -1, neither of above → set to 0. Switch may be preset to +1 if no symbol is needed for the current light button action.

Figure 4. Subroutine DPDEF, continued
LIGHT BUTTONS


DESTROY —— Print error comment. (This light button should have been processed above.) EXIT

ESCAPE —— Identify symbol of expression indicated (subroutine EXLOC). Blank symbol mark. Display expression. Set switch=0

REPLACE SYMBOL —— Replace symbol name in temporary expression pack with symbol name from list.

MOVE SYMBOL UP/DOWN/LEFT/RIGHT —— Identify symbol of expression hit. Calculate new X/Y coordinate. Display expression. Set switch=1

ERASE ——

Figure 4. Subroutine DPDEF, continued
Figure 4. Subroutine DPDEF, concluded
Figure 5. Display Screen for DFDEF
Ready display, blank screen.

Find and confirm current menu name. Enter menu name in MSYMTBL list and create menu list or obtain pointer to existing menu list.

Call CAMSET which executes PDP-8 code to save main display file locations.

Transmit a square as a subroutine display file. Transmit a small square as a main display file. This will later be used to indicate the Grafacon location by altering the end point of the first invisible vector.
Transmit the light buttons, each as a main display file containing a push-jump to the subroutine box, the text of the label, and a stop blink code.
Transmit the grid of endpoints by first generating and transmitting as a subroutine file a line of eight points. The grid is displayed as two main display files each containing four push-jumps to this subroutine file.
Transmit a box to contain the symbol. The user generates the symbol on a large scale for convenience. The symbol in the box shows the actual transmitted size.

Call CAMSET to stop saving main display file locations.

Find the data pack containing the PDP-8 symbol generation program.
Call PRØG to transmit the program to the PDP-8. PRØG executes a PDP-8 task which reads the transmitted program, stores it, and transfers to the starting location of the transmitted code.

Figure 6. Subroutine DCSYM.
Figure 7. Display Screen for DCSYM
Ready display, blank screen.
Set LMP=0.

Display light buttons. Draw separation line.

Display menu names contained in MSYMTBL above light buttons.
Draw 2 lines above menu names.

Reserve DF numbers.

Find and confirm current problem name.
Enter name in NAMEFLAG table.
Enter name in MPRØB list and create association table.

Transmit symbols which already exist in this problem as invisible subroutine files (subroutine I8REG).
Create a temporary data pack. Move problem name into negative region of pack.
Move problem symbols into temporary pack as entries in an expression pack.
The coordinates position the symbols in the problem symbol display area.
Display the symbols.

Create a temporary data pack.
Move problem name to negative region of pack.

Figure 8. Subroutine DSYSEL
Blank light button mark.
Request a Grafacon hit on a light button.

Locate and mark the light button hit.

all light buttons except ADD ÆNE
SYMBOL TO PROBLEM

Blank menu symbols displayed.

LMP=0

Free space used for menu names and pack pointers.
Set LMP=0.

Process the light button.

Figure 8. Subroutine DSYSEL, continued.
LIGHT BUTTONS

DISPLAY CONTENTS OF MENU
or
ADD ENTIRE MENU TO PROBLEM

- Request Grafacon hit.
- hit above light buttons
- Determine if menu name hit.

Identify menu name.
Get space and find all symbol names
and corresponding data pack pointers
Branch for specific light button.
for indicated menu.
Set LMP ≠ 0.

Create a temporary DF data pack.
Find pointer to pack containing display file
for first symbol.
Move DF header from this pack into temporary DF pack.

Convert Y and X coordinate to correct form for display file.
Put escape bit on X.
Move PAR = 0417, MODE = 1127,
Y, X, PAR = 0617 into temporary DF pack.

Move symbol display file to temporary DF pack omitting header and PAR at front and stop blink or PAR at end.

Have all symbols been processed?

Increment X and Y.

Calculate word 7 of the DF header and store over previous value in temporary DF pack. Set byte 1 of word 1 of the DF header in the pack to 40\text{16}.

Transmit the pack as an invisible display file (subroutine DFPTB).
Generate a main display file to display the subroutine file.
Free space used for temporary DF pack.

Figure 8. Subroutine DSYSEL, continued
ADD ENTIRE MENU TO PROBLEM

Is (current number of symbols + number of symbols in menu) > 63?

yes → Print comment that symbols will not fit in problem.

no → Set switch=1

Find symbol name. Enter association in problem association table.
Move symbol name into temporary pack as an entry into an expression pack.

Find pointer to pack containing display file for symbol.
Transmit symbol as invisible subroutine file.
Move DF number into negative region of symbol display file pack.

Are all symbols processed?

yes → Display all problem symbols.

no →

LMP ≠ 0 → Free space used for menu names and pack pointers.
LMP = 0 → Free space used for temporary pack.
Save MPRØB list and problem association table.

ESCAPE → Blank screen.

Figure 8. Subroutine DSYSEL, continued
**ADD ONE SYMBOL TO PROBLEM**

- Is LMP = 0?
  - yes → α
  - no → β
- Request Grafacon hit.
- hit above light button.

**DELETE ONE/ALL SYMBOLS FROM PROBLEM**

- Determine if symbol of displayed menu hit.
  - yes → Set switch=2 → Display all problem symbols.
  - no → γ
- Empty temporary expression pack and problem association table.
- Blank problem symbols.
- Free DF numbers of invisible subroutine files.

**Figure 8. Subroutine DSYSEL, concluded**
Figure 9. Display screen for DYSSEL

<table>
<thead>
<tr>
<th>LIGHT BUTTON AREA</th>
</tr>
</thead>
<tbody>
<tr>
<td>MENU NAMES DISPLAY AREA</td>
</tr>
<tr>
<td>MENU SYMBOLS DISPLAY AREA</td>
</tr>
<tr>
<td>PROBLEM SYMBOLS DISPLAY AREA</td>
</tr>
</tbody>
</table>
2.3 SUBROUTINES USED IN DPRØB COMMAND.

Subroutine SE8 is used to display a string of symbols contained in a data pack with entries as shown in Fig. 2. The string could be an expression, entries in a matrix, or a list. The subroutine has one entry point, SE8MY, which can be used to set the minimum Y value to be displayed. This prevents the string from displaying over light buttons or other information located at the bottom of the screen. The execution of SE8 and SE8MY is outlined in Fig. 10.

Subroutine EXØRD orders one expression data pack into another with increasing X, then increasing Y as outlined in Fig. 11.

Subroutine EXLØC examines an expression data pack and returns the displacement to the entry located at a specified coordinate position (see Fig. 12). The entries in the data pack are not in any particular order.

Subroutine INITØ locates and transmits all the symbols of a problem as invisible display files. The DF numbers are acquired dynamically and then saved for reference in the negative region of the symbol display file data pack. The entry point IØREG also saves the DF numbers in an array with the array subscript equivalent to the position in the problem association table. The execution of INITØ and IØREG is outlined in Fig. 13.
Arguments:
LPK = pointer to expression data pack,
NUM = display file number for expression.

Retrieve problem name from negative region of pack.
Find pointer to problem association table.

Blank display file NUM.
Initialize display file.

Read entry from LPK.

Are values of X and Y coordinates within display area?

Find name of menu containing this symbol from problem association table.
Find pointer to menu name list.
Find pointer to symbol display file data pack in menu name list.
Find invisible subroutine display file number in negative region of symbol display file data pack.

Enter Y, X with escape, PAR to reset the scale, a push-jump to the symbol subroutine file, and standard PAR and MODE into the expression display file.
Reset the internal DF light-beam coordinates.

Entry SE8MY in subroutine SE8.

Argument:
MY = minimum Y for display area.

Set minimum Y to be used for display area lower bound in SE8.

Figure 10. Subroutine SE8.
Arguments:
LEXP = pack to be ordered,
LEPK = pack to contain the result of ordering.

Empty pack LEPK.
Set displacement for pack LEPK to 0.

α

Read entry from LEXP.

Is scale ≥ 0?

no

yes

Save displacement to this entry and X and Y values.

β

Read next entry in pack.

Is scale ≥ 0?

no

yes

If (new X < saved X) or (new X = saved X and new Y < saved Y), replace saved value with corresponding values for this entry.

γ

Copy pack LEPK to pack LEXP.

β

Move entry with lowest X,Y to pack LEPK.
Increment displacement for LEPK.
Set scale in entry in LEXP to -1.

α

return

Figure 11. Subroutine EXORD
Arguments:
IXL, IYL = position usually of Grafacon hit,
LEXP = expression data pack,
KDIS = displacement.

Read entry from pack.

Does IXL, IYL fall within symbol? (The entry contains
the minimum X,Y for the symbol. The upper limits are
calculated using the scale factor of the symbol.)

no

Have all entries been examined?

no

yes → return with displacement
to symbol.

yes

IXL, IYL did not fall
on a symbol.
return 1

Figure 12. Subroutine EXLØC
Arguments:
SPRØ = problem name.
NSUB = array to save subroutine display file numbers.

INIT8
- Set switch=1
  - Find pointer to problem association table. Find all symbol and corresponding menu names in problem.
  - Find pointer to pack containing display file for symbol. Transmit symbol as invisible subroutine file. If switch=2, save DF number in array NSUB. Move DF number into negative region of symbol display pack.
- Are all symbols processed?
  - no
    - return
  - yes
    - return

I8REG
- Set switch=2

Figure 13. Subroutine INIT8
Subroutine FONE is used to define and mark entities. The user indicates the position of the entity marks--above or below the expression--and points to the first and last symbols to be included in the entity. The expression is blanked and redisplayed with the indicated group of symbols marked.

The subroutine exists as a routine independent of CAMA in file FONE with changes necessary to task under CAMA indicated on the listing. A test program is located in FONE (-100,-97). The object of the subroutine only is in FONEO. The RUN command is
RUN (test)+FONEO+MATH+SAVE:XDF+SAVE:CAMAL 2=SS73:TSAV2.
The execution sequence of the subroutine is outlined in Fig. 14. The display screen is illustrated in Fig. 15.
Ready display, blank screen.
Display light buttons.

\( \alpha \)

Find and confirm current problem name.
Enter name in NAMEFLAG table.

\( \beta \)

Find and confirm current expression name.
Create a temporary data pack.
Move expression into temporary pack.
Display expression.

Wait for a Grafacon hit on a light button.

\( \gamma \)

Locate and mark light button hit.
Process the light button.

Figure 14. Subroutine EXØNE
LIGHT BUTTONS

DEFINE LOWER/UPPER ENTITY → Determine entity number. → Request Grafacon hit 1. → hit on expression → Find location of symbol 1 hit in data pack. → hit on light button

1 → Request Grafacon hit 2. → hit on expression → Find location of symbol 2 hit in data pack. → Create temporary pack for expressions with entity marks. → Search entity for minimum/maximum Y. The entity marks are displaced 7 raster points from this value.

2 → Move first part of expression to new pack, insert left entity mark with entity number in entry, move entity with increased X, insert right entity mark with entity number in entry, move remainder of expression with increased X. → Free space of old temporary data pack. → Display new expression with marks. Increase entity number.

CHANGE PROBLEM
or
CHANGE EXPRESSION

CHANGE PROBLEM → Free space of temporary pack. Blank expressions, light button mark.

CHANGE EXPRESSION

ESCAPE → Blank display. → Free space of temporary pack. → return or Task DMAIN

Figure 14. Subroutine EXONE, concluded
Figure 15. Display Screen for FONE

The Computer-Aided Mathematical Analysis system (CAMA) permits construction of symbols and expressions, manipulation of expressions, and solution of equations. CAMA operates under MTS in the IBM 360/67 and RAMP in the PDP-8/338 terminal. CAMA operations are divided into tasks which may be inserted into the queue via commands.

The command DPRØB is used to define a problem. The user selects symbols to use in the problem, constructs symbols if necessary, and formulates the expressions which comprise the problem.
<table>
<thead>
<tr>
<th>KEY WORDS</th>
<th>LINK A</th>
<th>LINK B</th>
<th>LINK C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ROLE</td>
<td>WT</td>
<td>ROLE</td>
</tr>
<tr>
<td>define problem</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>data structure</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>display</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>expression</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>symbol</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>