MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Artificial Intelligence
Memo No. 182

March 1970

Display Functions in LISP

Thomas O. Binford

This note describes a system which compiles various forms of LISP lists and arrays into display commands for the DEC 340 display, and provides supporting functions for scaling, for moving elements in a display, for pot control of certain displays, and for adding elements to and removing elements from the display.

Display Functions in LISP

INTRODUCTION

In the examples given, arguments in lower case are dummy arguments; the user's arguments are to be substituted. The symbol ^ preceding a character signifies a control character, and _ signifies the character "space".

An EXPR subset of this system is available, accouate for small displays. A SUBR version of these display functions, written in MIDAS with control routines in interpreted LISP, is available in a special version of LISP called DLISP. As display primitives, the system compiles points, open and closed curves, and text. Functions handle input lists and arrays in a variety of forms. Special case primitives such as coordinate axes, box, and cross are useful for making graphs and figures for reports.

The system provides a scaling and translation (no rotation) from a rectangle in the domain to a rectangle in the range, at the time of compiling. Only very weak scaling is available in the hardware (vectors and text scale by powers of two); the best solution is to re-compile the list with revised scaling. Several functions aid in choosing scale. Most displays can be translated within the limits of the device, either with absolute coordinates, or under pot control.

Individual elements of a display can be added or removed. Whole displays can be saved, copied, and restored. A convenience feature is that removal of elements can be done by blinking each element in turn, and interrogating the teletype.

The system depends heavily on dynamic array allocation and automatic garbage collection of arrays. LISP has been modified to include this feature. In normal operation of the display system, elements which are not being displayed or saved, or otherwise protected, are garbage collected. The user may provide the array in some cases, or the name of the array. The system normally provides necessary arrays as a convenience. A function for generating mnemonic atoms is provided. Except when the user provides the array, exact lengths of displays are calculated before array allocations.

OPERATING THE DISPLAY

There are only two steps in initiating a display:

1) type ^F to select the display device,
2) make a function call to create a display array.

This section aims to sketch out the general classes of functions which are described in detail in later sections. The names of all functions in the display system begin with the letter D. The commands to turn on or turn off the display are control characters which can be typed at any time to EVAL (even in the middle of an atom). These commands are handled in interrupt mode and take effect immediately, so that they can be used even when LISP is running.

With display output selected, elements are displayed as soon as they are created. These are really two operations, compiling display commands (in a LISP array) for the scope, and adding the created array to the list of arrays on display. Two lists contain the elements of the display. DISPLAY contains the atoms whose arrays are on display (for the benefit of humans) and DISLIST contains their array properties, which the ITS system uses for display.

		DISPLAY MODE	TEXT MODE
1.	To turn on the display	^F	^N
	To turn off the display	^Y	^Y

2. To create display arrays and add to the arrays being displayed, the function used is specified by the type of display desired (point, vector, or text) and the type of input data (list or array of numbers or dotted pairs or atoms which evaluate to numbers or dotted pairs). The following is a list, without details, of available functions.

DISPLAY TYPE	FUNCTIONS
TEXT	DTEXT
	DTEXTE
OPEN CURVE	DCRV
	DLV1,DLV2
	DAV1,DAV2
	DA2V1
CLOSED CURVE	DCRVC
	DLV1,DLV2
	DAV1,DAV2
	DA2V1
POINT	DPTL
	DLP1,DLP2
	DAP1,DAP2
BOX	DBOX
	DGRID (under pot control)
AXES	DAXES
CROSS	DCROSS
	DPTS (under pot control)

3.These functions manipulate displays, adding elements to or deleting elements from a display, or saving and restoring displays.

DISPLAY, DISPLAYE
DCLEAR, DCLEARE
DCOPY
DSAVE
DGET

Another class of functions consists of those which move, or manipulate individual elements of a display.

DMOVE
DPTS
DBRIGHT

POINT AND VECTOR DISPLAYS

What basic displays might be useful? We might want to display a set of points, or a curve made of straight line segments connecting points. A usual form of display input might be a list of dotted pairs of x and y coordinates, or a list of such lists. For this usual form there are three basic functions:

```
DPTL    EXPR    displays a set of points,
DCRV    EXPR    displays open curves,
DCRVC   EXPR    displays closed curves.
```

For example, (DPTL list) or (DCRV list) are typical uses.

We might want axes, a box, or a cross. The examples are:

```
(DAXES (LIST (a1 . a2) (u1 . u2)))
(DBOX (LIST (a1 . a2) (u1 . u2)))
(DCROSS  name (x . y) size)
```

where (a1 . a2) is the lower left corner of a rectangle aligned along the coordinate axes and (u1 . u2) is the upper right corner.

There are a number of other input options for which functions exist in the SUBR version. We could have list or array input. We might want to graph a set of values (numbers) evenly spaced along an axis. In that case, the x coordinate would be calculated by scaling the indices 0,1,2, . . n. The functions are named by acronyms which correspond to the options:

```
A or L  list or array input
P or V  point or vector (curve) display
1 or 2  numbers (graph) or dotted pair input.
```

List input may contain atoms which evaluate to dotted pairs, provided

the flag DATOM is non-nil. The flag should otherwise be nil to avoid the associated cost of this feature.

LSUBR

DLV1	Display List of Vectors (numbers)
DLV2	Display List of Vectors (dotted pairs)
DLP1	Display List of Points (numbers)
DLP2	Display List of Points (dotted pairs)
DAV1	Display Array of Vectors (numbers)
DAV2	Display Array of Vectors (x,,y pairs)
DAP1	Display Array of Points (numbers)
DAP2	Display Array of Points (x,,y pairs)

SUBR

DA2V1	Display 2-dim Array of Vectors (numbers)

Some examples of calls are:

```
(DLV1 list)
(DLV1 list 'array1)

(DAV1 arrayin arrayout min max)
(DAV1 arrayin max)
(DAV1 arrayin)
(DAV1 arrayin min max).
```

Arrayout, min, and max are optional arguments. Min and max are least and greatest indices for the input array. If min is given, max must be given. The default cases for min and max are the limits of the array.

With the functions for list input, DLV1, DLV2, DLP1, and DLP2, closed curves can be generated by the calls (DLV1 (DCC list)) and likewise for DLV2, DLP1, and DLP2. At present, no provision is made for closed curves for those functions with array input.

A typical call for DA2V1 would be:

```
(DA2V1 '(arrayin) (a1 . a2) (n1 . n2) ((u1 . u2) . (v1 . v2)))
```

over the two-dimensional grid given by:

```
x0=a1+n2*v1 ;outer loop along vector (v1 . v2)
y0=a2+n2*v2
x=x0+n1*u1 ;inner loop along vector (u1 .u2)
y=y0+n1*u2.
```

SCALING

Points are scaled if DSCALE is non-nil. The two coordinates may be floating point or fixed point LISP numbers. If DAIOM is non-nil, non-numeric atoms are evaluated to dotted pairs. Scaling and display are a mapping from some two-dimensional rectangular DOMAIN into the two-dimensional rectangular RANGE (scope coordinates). The limits of scope coordinates are (0 . 0) to (1777 . 1777).

Scaling corresponds to the following operations:

```
DSCALE  =       (sx . sy) ;scale
DORGD   =       (xd . yd) ;origin of the DOMAIN
DORGR   =       (xr . yr) ;origin of the RANGE
u=sx(x-xd)+xr
v=sy(y-yd)+yr.
```

The free variables which control the scaling are:

DOMAIN	RANGE	
DORGD	DORGR	origin of mapping
DLOWD	DLOWR	lower left corner
DUPD	DUPR	upper right corner
	DSCALE	

Two utility functions aid in determining scale:

DOMAINL EXPR determines the extreme coordinates of a list of dotted pairs.

DSCALE EXPR determines the scale, given domain and range variables.

FUNCTIONS FOR MANIPULATING DISPLAY LISTS

We want to be able to add elements to or delete elements from DISPLAY and DISLIST, to save, copy, and restore display lists. A slight complication should be kept in mind: most arrays are atoms which are not on the OBLIST. They cannot be referred to directly by naming them, but must be referred to by something which points to them. That is most troublesome in deleting elements. A special mode in DCLEAR blinks each element in turn, preserving or deleting as instructed by the user in a teletype control loop.

DCLEAR FEXPR
(DCLEAR) removes all arrays
(DCLEAR a b c) removes named arrays
(DCLEAR TTY) proceeds down the list DISPLAY, blinking the current array. Teletype commands are:
Q_ quit
T_ proceed and preserve the array
K_K_ kill the current array.

DCLEARE EXPR evaluates its argument and removes a single array.

DISPLAY FEXPR
(DISPLAY a b c) adds the atoms to DISPLAY and their arrays to DISLIST.

DISPLAY EXPR
evaluates its argument and adds a single array.

DCOPY FEXPR
(DCOPY world)
(DCOPY)
conses a copy of DISPLAY onto DSAVE with the scene name world (or a GENSYM if no name is given).

DSAVE FEXPR
(DSAVE world)
(DSAVE) is equivalent to (DCOPY)(DCLEAR).

DGET FEXPR
(DGET world)
(DGET)
appends the saved scene (or all scenes if no name is given) to DISPLAY.

DISPLAYING TEXT

The simplest mode of text display uses the display like a teletype, as a character output device. Material printed goes to the scope as to a teletype. The control characters ^N and ^Y turn text output on and off, respectively. Text mode and display mode are incompatible; to label displays, other modes must be used.

The DTEXT display call specifies a line of text (provides for no carriage return), with optional argument for vertical alignment, and optional absolute position. Pot control (pots 146,147) is used in absence of an absolute position.

DTEXT FEXPR
```
(DTEXT (line of text) Y p)
(DTEXT (line of text) p)
(DTEXT (line of text))
(DTEXT (line of text) Y)
```
If the second argument is Y, the text is displayed vertically, if not, the second argument is taken as a position (dotted pair p). If there is no position specified, the text is put under control of pots 146 and 147; T_ terminates pot control.

DTEXTE EXPR
```
(DTEXTE '(line of text) T p)
(DTEXTE '(line of text) NIL NIL)
```
This form evals its arguments; if the second argument is non-nil, the line is displayed vertically. If the third argument is non-nil, it is taken as a position, otherwise position is under pot control as in DTEXT.

The other means of text display is compatible with point and vector displays and provides carriage return as in PRINT. The first step is to allocate and initialize the array for text:

```
(ARRAY name NIL length)
(DISINI 'name)
```

Text can be added incrementally by:

```
(DISAD 'name flag 'abc)
```

The second argument should be T or NIL depending on whether PRINI or PRINC type output is desired. Positioning is left to the user. The byte pointer is stored in the last word of the array.

DISINI SUBR
(DISINI 'name)
initializes the array for character mode output.

DISAD SUBR
(DISAD 'name flag '(text to be added))
Appends the s-expression third argument to the text in the array name in the print mode specified by the second argument.

DISCNT (display character count) functions as CHRCT does for PRINT.

OUTPUT TO THE PLOTTER

Any display can be plotted with the command (DPLOT), which turns the paper and plots the current display. Thus, the whole display apparatus is available for plotting.

for special uses, complete control of the plotter is available with the commands (PLOT n), where the argument n indicates a character, pen up, pen down, etc. (CHAR PLOT, AI memo 125, M. S. Speciner; THE CALCOMP PLOTTER IN TS AND LISP, AI memo 136, M. S. Speciner).

MOVING DISPLAYS AND POT CONTROL

To move an array with a point position, execute:

(DMOVE x y dname)

where dname is the name of a display object.

To determine a rectangle (aligned with the x-y axes) in scope coordinates under pot control, we execute (DGRID)T_ where T_ terminates.

We can move most arrays (but not all arrays) under pot control by using (DPTS a) where a is an array. If we call (DPTS NIL) we control a cross on the screen. We can locate points in the display by typing in names to DPTS which marks the spots with crosses, named after the atoms typed in, and sets the values of the atoms to the location in scope coordinates. Such a usage would be:

(DPTS NIL)a_b_c_T_

where T_ terminates.

DPTS EXPR
moves certain display arrays on the scope, under control of pots 146 and 147. If the argument is nil, the array moved is a cross. The following s-expressions are commands:

T_ terminate and return
FINE_ enter a fine control mode centering aroung the current location
COARSE_ resume the coarse position mode; the point may shift drastically for obvious reasons.
atom_ the atom is given an array, into which a cross is put, and whose value is the current location (dotted pair). The cross is left at the current location.

UTILITY FUNCTIONS IN DLISP

DLISP contains some standard utility functions:

SIN	arg in radians, $10^{-8} < x < 10^{8}$
COS	same
ATAN	(ATAN Y X) computes arctan y/x
SQRT	
EXP	exp base e; arg $-83 < x < 83$
LOG	log base e; arg $10^{-38} < x < 10^{38}$

EXAMPLES

The following is an example of a program which uses the display to show the successive stages of analysis:

```
(IOC F)
(EDGES)
(COND (SHOW (DCLEAR) (SHOWENDS)
(DTEXTE '(ENDS) NIL UPPT)))
(COND (PLOT (DPLOT)))
```

where the function SHOWENDS is defined as:

```
(DEFPROP SHOWENDS (LAMBDA () (PROG (A B)
(MAPC (FUNCTION (LAMBDA (X)
(COND ((SETQ A (GET X (QUOTE CORNERS)))
(SETQ B (CONS A B))))
                                )) SEGMENTS)
(DCRV B)
        )) EXPR).
```

AVAILABILITY

The two forms are available as:

EXPR (UREAD DIS EXPR COM)

SUBR DLISP^H
(UREAD DIS SUBR COM).

Further implementation is invited.

Some changes and simplifications are planned. Revised versions of this description are to be found on the user directory .INFO. on the ITS system.