Contents of Page 466-467 of:

Wiseman, N.E., Lemke, H.U. & Hiles, J.O. (1969) PIXIE: A New Approach to Graphical Man-machine Communication, *Proceedings of 1969 CAD Conference Southampton*, IEEE Conference Publication 51, 463 – 471.

Transcribed from the original
William Buxton
July 7, 2008.

Note: I have transcribed the section of the article dealing with interaction, rather than scan the photocopy, since the bleed-through from the other side of the page makes the copy difficult to read.

The transcription begins at the start of pgh 2 of page 466:

…

The user's actions comprise typing commands and messages on the teletype, pointing the lightpen at lightbuttons on the screen and training a tracking cross around the screen by moving the lightpen over it. Two sets of lightbuttons are provided:

1. The command lightbuttons which are for inducing gross actions such as changing mode displayed down the right edge of the screen.
2. The control lightbuttons which are used frequently within a mode to carry out a sequence of operations. These buttons are displayed around the tracking cross and move about with it so that the user's hand is always close to these buttons when he needs them. To avoid clustering a large number of control buttons around the tracking cross, it is arranged that only buttons for those actions which are legal at any given time are displayed and also that the user may select different sets of legal buttons by pointing at one of them (which acts as a sort of rotating switch for the rest).

To illustrate an action on the screen consider how the user might draw the fragment of circuits shown in figure 2.

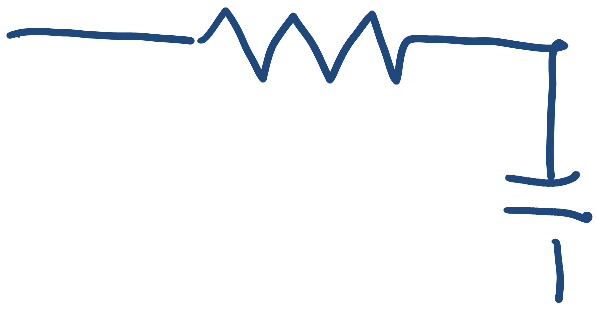

FIGURE 2 (hand-drawn reproduction by Buxton)

He points the lightpen at command buttons EN (end previous work, make clean start); DR (enter drawing mode); and ST (set constraints on line-drawing to be straight horizontals or verticals). Next he picks up the tracking cross and moves it to a position on the screen where he wants to start drawing and points at the control button S (start drawing). When he ext tracks the cross, in say roughly W->E direction, the graphical communication routine constructs segments of line in the temporary display file in exactly W->E direction (since horizontal-vertical constraints are selected) and displays them on the screen. When he wants to introduce a symbol he selects the appropriate control lightbutton (in this case R for resistor) and the symbol is inserted in the display file in the appropriate orientation. As he continues to track, further line segments are added, changing direction (in this case to vertical) when his hand motion strays by more than about a quarter inch from the line being drawn. Next he selects a capacitor by pointing at the control button C: note that it is inserted in vertical orientation because he is drawing vertically [page break to page 467 of original text] now – and finally draws a further short segment of line and finishes by pointing at the button F. Up to this time he has been able to erase and correct the drawing simply by retracing the pen over its path (a technique we call "undrawing"). The graphical communication program dost this simply be maintaining a stack recording the items to be put in the temporary display file and pushing or popping the stack according to the motion of the pen.

When the F control button is hit, radical changes in the format of data in core occur, although the picture on the screen remains unchanged. The up-compiler (2) translates the information from the temporary display into a structured form representing the nods and branches, lengths and positions of line segments and so on and appends it to the existing data structure (unless of course, as is the case for the example above, there is no existing structure, in which case a new one is started). The up-compiler has display file as source and RSP data structure as object and is primarily concerned with extracting features about the circuit from what is in the temporary display file and the existing data structure. It is at this point for example that notions of connectivity first enter into the format of the data being put together in the machine. We shall see later how the user's actions make reference to such features. After the amended data structure is formed the down compiler (3) is entered to generate a new display file (the "permanent display file") after which the temporary display file is emptied. The drawing actions may then be recommenced. Information in the data structure is store in a hierarchical form (circuits, branches and nodes, squiggly lines) and the permanent display file is linked with the data structure at the various levels to assist in the identification of pen hits from the picture. This is the purpose of the permanent display file and, until some action arises which needs such identification aids, the appearance of the picture is unchanged by transitions from temporary to permanent status.

Suppose next that the program enters pointing mode (user points at the command button PO). The menu of control buttons now changes to display the options available in this mode, such as T (for track) and E (for erase). When a hit from the permanent display file occurs the context of the hit is displayed b causing all or part of the picture to blink on and off at about twice per second. The context can be moved up and down in the hierarchy by hitting further command buttons until the desired picture part is blinking. For example, a line, or node which contains the line, or a subcircuit incorporating that node can be identified, and made to blink, when the line is pointed at and this picture part can then be the operand for a further function. Examples of functions available are

1. Track the picture part around on the screen (point at T and track the cross).
2. Erase the picture part (point at E)
3. Give the name or lable to the picture part (type it in on the teletype.)

4. Attach arbitrar formatted messages to the picture part (type themin with the particular prefix characters)
5. List all the data about the picture part on the teletype (type a space).