DOCUMENT RESUME

ED 030 458 LI 001 537

By-Haynes, Leonard Stanley

Video Communication Program.

Pennsylvania Univ., Philadelphia. Moore School of Electrical Engineering.

Spons Agency-National Science Foundation, Washington, D.C.

Pub Date May 69

Note-29p.: Thesis, Moore School of Electrical Engineering, University of Pennsylvania, 1969.

EDRS Price MF-$0.25 HC-$1.55

Descriptors-*Display Systems, *Documentation, *Indexing, *Information Retrieval, *Man Machine Systems, Subject Index Terms

Identifiers-VICI, Video Console Indexing

This thesis describes work done as part of the Video Console Indexing Project (VICI), a program to improve the quality and reduce the time and work involved in indexing documents. The objective of the work described was to design a video terminal system which could be connected to a main computer to provide rapid natural communication between the user and the system. The report discusses the nature of the communication which an indexer may require, such as the definition of a word, terms generic to a given term, synonyms of a term, abstracts of a document, etc. The video terminal must have the following capabilities: (1) to permit the user to make the above requests in a rapid, natural manner and send the appropriate code to the main computer; and (2) to receive the main computer's response to any request and display it on the screen. The indexer should be able to work directly from the screen. Since the screen can display only a very limited number of characters, the video terminal must contain a set of instructions to allow the user to control what is visible on the screen. The hardware and software for the system are described. (Author/CC)

VIDEO COMMUNICATION PROGRAM

by

Leonard Stanley Haynes

May 1969

UNIVERSITY of PENNSYLVANIA

The Moore School of Electrical Engineering

PHILADELPHIA, PENNSYLVANIA 19104

University of Pennsylvania
THE MOORE SCHOOL OF ELECTRICAL ENGINEERING
Philadelphia, Pennsylvania

VIDEO COMMUNICATION PROGRAM

by

Leonard Stanley Haynes

May 1969

The work described in this thesis was fully supported by the National Science Foundation under a study entitled "Real-Time Video Console Indexing", Grant No. GN-682.

The Moore School Information
Systems Laboratory
University of Pennsylvania

ACKNOWLEDGEMENTS

The research involved in this work was done under the direction of Dr. Morris Rubinoff, and grateful appreciation is expressed for his many suggestions and invaluable guidance.

Dr. Warren Seider was also a source of valuable suggestions, and many of his ideas are incorporated in the system operation.

Finally, the entire student staff of Project VICI all made valuable contributions, and I would like to extend my thanks to them.

TABLE OF CONTENTS

1. BACKGROUND

This thesis describes work done as part of Project VICI (Video Console Indexing) at the University of Pennsylvania. VICI is a program to improve the quality and reduce the time and work involved in indexing documents. In connection with this project, the objective of the work described was to design a video terminal system which could be connected to a main computer to provide rapid natural communication between the user and the system. The VICI proposal discusses the nature of the communication which an indexer may require. He may, for example, request the definition of a word, terms generic to a given term, terms specific to a given term, synonyms of a given term, words which are related to a given word by a cause-effect relationship, words related to a given word by a whole-part relationship, or the abstract of a particular document. The video terminal, therefore, must have the following capabilities:

1) It must permit the user to make the above requests in a rapid, natural manner and must send the appropriate code to the main computer.
2) It must receive the main computer's response to any request and display it on the screen.

The document indexer should be able to work directly from the screen; hence he should be able to look back at his previous work, compare various main system responses such as lists of synonyms and terms specific and generic to a given term, etc. Since the screen can display only a very limited number of characters, the video terminal must contain a set of instructions which allow the user to control what is visible on the screen in a manner versatile enough for him to work efficiently with as little "overhead" as possible.

2. I/O SYSTEM

2.1 Hardware

To implement a practical video terminal, the video device used should:

1) be inexpensive
2) be designed so that many terminals can be connected through dataphones to either a satellite computer or a main computer
3) be able to display text on a screen in a manner which requires little or no maintenance by the main computer.

There are many devices presently available off the shelf which might have been used. Two such devices were available at the Moore School. These are the DEC 338 and the Sanders 720. The DEC 338 costs about ten times the price of a Sanders 720 for each screen and is much less useful for the project purposes since it can put only 250 characters on the screen without unacceptable flickering, while the 720 can display 1024 characters with no visible flickering. Hence, the Sanders 720 was the obvious choice.

Each 720 terminal contains its own 1024 word delay line memory. The screen size is seven by nine inches, which is enough for 2048 characters to fit on the screen. (This is not incompatible with the memory size since consecutive spaces need not be stored in memory.) The 720 also has hardware editing capabilities; hence no software need be written to aid the user in preparing input for the system. Table 1 is a complete list of the characteristics of the Sanders 720.

2.2 Software

A video output system has its main advantage in its high rate of information output. However, it has a major disadvantage in that a user cannot look back at previous work without requesting the system to put the information back on the screen. In fact, numerous common operations, such as having certain past and present output images visible simultaneously will not even be possible without specific commands which perform the required image manipulations. Deciding which capabilities to implement in the VCP was the first problem faced in designing the video I/O system. Note, in the discussion following, the word "interaction" is defined as the main information source's response to any user request. For example, the request for the definition of a word will result in an interaction. The interaction as far as the I/O terminal is concerned is simply the system's response to the querry, specifically the definition of the given word. The following list was chosen as the set of options which, if implemented, could be used effectively by the user.

1) The I/O system must provide the ability to "roll" the image on the screen so if not all the text is visible at one time, the rest can be brought into view.

TABLE 1

SYSTEM SPECIFICATIONS

Characters per Display:	1024, 512, or 256 are available
Characters per line:	
Vertical Displays:	52
Horizontal Displays:	64
Lines per Page:	
Vertical Displays:	40
Horizontal Displays:	32
Character Repertoire:	64 ASCII alphanumerics
Character/Function Code:	ASCII Standard
Viewing Area:	7½ x 9½ inches vertical or horizontal
Refresh Rate:	46.5 CPS
Character Height:	0.12 inch nominal
Character Width:	0.09 inch nominal
Character to Character Spacing:	0.05 inch nominal
Line to Line Spacing:	0.08 inch nominal
Character Generation Method:	synchronous, continuous strokes
Character Write Time:	21 microseconds
Deflection Method:	magnetic
Spot Size:	0.020 inch max.
Brightness:	30 foot lamberts min. (with a 6"x8" raster)
CRT Filter:	gray, 49% light transmission min.
Phosphor:	P4, type W modified
Storage Method:	recirculating magneto-strictive delay line
Communications:	
Control Unit to Display:	up to 1000 cable feet
Keyboard to Display:	up to 10 cable feet; ASCII code
Control Unit to Computer:	direct or remote; ASCII code standard
Keyboard:	attached or remote
Max. Serial I/O Rate:	5 characters every 21.5 milliseconds
Max. Parallel I/O Rate:	47.5 characters per second
Parity:	inserted in I/O Logic before transmission
Input Power:	
Display Unit:	115 VAC $\pm$ 10% @ approximately 200 watts, 60 CPS
Control Unit:	115 VAC $\pm$ 10% @ approximately 150 watts, 60 CPS
Keyboard Unit:	+10 VDC unregulated and + 4.5 VDC (supplied by display unit)

2) A user may wish to look back at what he did several "interactions" ago; hence the system must provide the ability to recall previous output to the screen.

3) The user may wish to keep all or part of one or more interactions on the screen at all times and continue to interact with the system having incoming information fill the remaining space on the screen.

    (A) The user may wish to roll these interactions independently.

    (B) The user may wish to change the amount of the screen allocated to these interactions.

4) The user may wish to have information on the screen in different formats, i.e., lists and paragraphs.

5) The user may wish to choose certain words on the screen as operands for his next command. For example, he may want to find the definition of certain words in an abstract without having to type these words.

This list could certainly be extended. For example, the I/O system could provide the ability to store words specified by the user as a list and place the list on the screen by a single command, or it could provide the user with the ability to reposition information on the screen arbitrarily (put one list beside another instead of underneath). These additions, however, greatly increase the complexity of the system software and the listed options should be sufficient.

3. ADDRESSING OF INTERACTIONS

3.1 Problem

The problem of how to address past or present interactions is the most difficult and most important problem faced in the specifications of the VCP. As a simple example of what is meant by "addressing interactions," assume a document indexer requests the abstract of a document. This is displayed on the screen. Next the indexer asks for the definition of a term in the abstract. This is retrieved from the files and displayed on the screen. Now, in general, there will not be enough room on the screen for both the abstract and the definition. Assuming for the moment that the definition is placed on the screen in its entirety and only part of the abstract remains visible, how does the indexer command the system to let him see the rest of the abstract? If the system is to permit independent rolling of interactions on the screen, a simple ROLLUP command will not suffice.

3.2 Sequential Scheme

A possible solution is to number communications successively from the most recent. In the example given, the user would type ROLLUP 1, 10, which would roll the most recent communication (1), the number of lines specified (10). This scheme, however is inadequate since the user cannot be expected to remember how many interactions have occurred since the one he is interestedin.

Continuing with the example, assume the indexer proceeds with his work, asking for definitions, synonyms, etc. He then wishes to take another look at the abstract which is no longer visible. He must command the system in a manner such as RECALL N, where N is the number of interactions ago that the abstract was called. Trying to figure out what N would be is time consuming as well as frustrating. The scheme would be prohibitively cumbersome.

3.3 Modified Sequential Scheme

A modified sequential address scheme was also considered. With this scheme, interactions are addressed by their type, followed by how many of that type have occurred since the desired one. Disadvantages with this system are:

1) The terminal system becomes dependent on the main information system. The terminal must be told how to label an interaction and modularity is lost.
2) If a user does, on occasion, use the same command a number of times, the system becomes cumbersome.
3) It would be almost impossible for a user to browse through his previous work.

3.4 VCP Addressing Scheme

Assume a user wishes to recall a previous interaction to the screen. If this desire is to be possible by a direct command, the command must be of the form RECALL N where N is some specific address of a general form. It was felt, in designing the VCP, that there is no general addressing scheme which would be both useful and easy to use. In the VCP there only four possible addresses---A, B, C, and D. All commands to manipulate information on the screen refer to one of these four areas. An interaction can be assigned address A, B, C, and D by the user or by the main information system. For example, the user asks for the definition of a word and specifies it to be addressed A. He can then roll the screen image of the resulting interaction by typing ROLLUP A, N which rolls the screen image of the contents of address A by N lines. A roll command will not change the number of lines of A displayed on the screen, only which lines are displayed. Similarly, he may specify the next interaction be assigned address B and may then roll B (up or down in either case). He may change the amount of each address space visible on the screen by typing ALOCAT N,M,P,Q, which allocates N lines to A, M lines to B,P lines to C, and Q lines to D.

The most recent interactions, assigned addresses A,B, or C, can be allocated screen space or rolled in the most general manner. Information may also be addressed D, however address area D is also used to refer back to previous work not in A, B, or C. Whenever information enters buffers A or B, the previous contents are automatically transferred to D. Hence A and B can contain at most one interaction each. When information is entered into D, old information is not lost. Area D contains as much information as will fill the buffer entirely, and when it is full, only the least most recently used information is lost, and only enough to make room for the entering interaction. Hence, the user can refer back to previous interactions not presently in A,B, or C by rolling through D. When information is entered into area C, old information is lost and cannot be recalled without requesting retransmission from the main information source.

Each area has a maximum size. This information plus the characteristics just discussed are summarized in Table 2. Note finally that if no buffer is specified for an interaction, A is assumed. If no screen allocation is specified, the last specified allocation is used.

3.5 Advantages of the VCP Addressing Scheme

1) The VCP scheme of image manipulation permits the user to browse through previous work by rolling D. He may keep a particularly interesting or important interaction on the screen at all times and have incoming information automatically use the remaining screen area. He may roll through previous interactions in D, comparing them with A, B or C. Hence, while not completely general, the addressing scheme provides for all the options listed in Section 2 as software specifications.

2) The I/O system command set consists of only two types of commands plus the command S (Switch-renaming A as B and B as A). The two types are ROLLUP (or ROLLDN) and ALOCAT. Hence, to accomplish all the manipulations mentioned above, the user need never use more than these two instructions.

3) In the VCP system, there are only four possible addresses. It is assumed that the user can remember what he has most recently put in areas A, B, and C. Area D contains everything else. In fact, remembering what is in A, B, and C is aided by the fact that the content of these areas are somewhat content dependent. For example, the user would put something in C which he will not want to

reexamine after it is no longer visible, perhaps because it is an "aside" from his main work, or because he wants it always present, such as with an abstract which he is indexing. Areas A and B would generally be used for important interactions and can be kept chronologically by using the S (switch) command. For example, if the user always types S to switch A and B and puts all data into A, the most recent interaction put in either A or B is always in A.

TABLE 2

Buffer A

1. Can hold a maximum of 1024 characters.
2. Can be rolled independently of any other buffer.
3. Can hold one interaction.
4. Any communication into A causes the previous contents of A to be transferred to buffer D.
5. Can be switched logically with buffer B, i.e., A becomes B and B becomes A.

Buffer B

1. Can hold a maximum of 1024 characters.
2. Can be rolled independently of any other buffer.
3. Can hold one interaction.
4. Any communication into B causes the previous contents of B to be transferred to buffer D.

Buffer C

1. Can hold a maximum of 4032 characters.
2. Can be rolled independently of any other buffer.
3. Can hold one communication.

Buffer D

1. Can hold a maximum of 3072 characters.
2. Can be rolled independently of any other buffer.

TABLE 2 (continued)

3. Communications are entered into buffer D either automatically by the VCP or by the user or main source. When new data comes in A, the old A goes into D. The same is true when data enters B, i.e., the old B goes into D. Buffer C does not transfer its previous contents into D. Finally, D may obtain information directly from the system in the same manner as information sent to A, B, and D.

4) The VCP system has an important advantage in that it can be used in a very mechanical manner with the user specifying almost NO commands. The system, however, remains useful and general.

A. Every communication from the information sources can specify

(1) switch A and B

(2) put the incoming communication in A.

The result of this policy is that systems interactions move from A to B and then from B to D. For example, if five communications have taken place, the most recent is in A, the next most recent in B, and the remaining three in D. The user can roll A, his most recent interaction, and B, his second most recent interaction independently, and he has access to the remaining previous interactions by rolling D. Buffer C can be used as a special purpose interaction buffer, perhaps for an abstract that will be of continuing interest.

B. The simplest mode of operation is to leave the buffer unspecified. The result is that all communications go into A and move from A to D. The system will automatically try to display all of A and any remaining screen space is allocated to D and displayed. Therefore the screen is like a long roll with all previous communication on it, except for the present interaction which can be rolled independently. The user may, of course, specify address B or C if he desires. This scheme, and the previous one, have the advantage that the user need almost never concern himself with manipulating what is on the screen, yet the system is still useful and despite its simplicity from the user's point of view, quite general in what can be done.

5) The last advantage of the VCP system of addressing is that it greatly simplifies the software necessary to implenent the various commends. It also saves storage since nothing need be remembered except the data itself which is simply stored on disk buffers named A, B, C, and D.

4. ADDITIONAL CAPABILITIES

A final capability is contained in the VCP. A user may type any command he wishes to the main information source. The VCP will append to that commend any word on the screen which has an asterisk (*) at the beginning, end, or anywhere inside the word. For example, user 8 has asked for the definition of a word, say NAND. He does not understand some word in the definition of NAND which the system has put on his screen. He types in the function block DEF, and then moves the curser (using the move curser keys on the console) and puts an asterisk (*) next to any word or words which he desires to know the definition of. The VCP will append these words to the DEF command before transmitting the command to the main information source.

5. SCREEN LOADING POLICY

In the following discussion, the concept of address spaces A, B, C, and D will be identified by the more physically meaningful term buffer A, buffer B, etc.

The Sanders 720 has two major limitations on what can be displayed.

1) There is a memory limitation of a maximum of 1024 characters.

2) There is a screen size limitation of a maximum of 32 lines of 64 characters per line (or 2048 characters).

Because of these limitations, the job of loading the screen according to allocation register is quite complex. The user may ask for 20 lines of A and 12 lines of B, but even the 20 lines of A may not fit on the screen. A user may not know how many lines an incoming communication will be. Normally he will want to see all of the communication, so he will specify all the lines to the buffer the information is coming into. (This is done automatically if he makes no specification.) However, the interaction may be very short, and unless some automatic decision is made to decide what else, if anything, is to be displayed in addition to the short interaction, most of the screen will be wasted.

Another problem occurs in the following example. A user specifies he wants to see 20 lines of C which is the abstract of a document. He leaves this on the screen, and incoming interactions, perhaps synonyms of a word in the abstract go into A which is allocated 12 lines. If the VCP tried to put 20 lines of C on the screen, it would run out of characters and get none of A. Probably the optimum allocation in this case would be to use all 32 available lines with 14 lines of C and 18 lines of A. This is possible since A is a list and therefore much less "dense" than C.

It is the job of the output supervisor package to decide the questions discussed above. Thus, the output supervisor must decide what to do when:

1. there are not enough characters to fill the allocation request of the user or system;

2. there is not enough data to fill the user or system allocation request, or, in other words, more lines are allocated than there are lines in the interaction;

3. the data to be displayed is in different formats, and the most efficient display is a function of the density (or format) of the data.

The output supervisor is a two-pass processor. If a user specifies that he desires 20 lines of C and 12 lines of A, 20/32 * 1024 characters of C and 12/32 * 1024 characters of A are allocated for the first pass. Therefore 640 characters of C, or approximately 10 lines of full text are allocated. 384 characters of A are allocated, but because of the low density of A, all the allocated lines are put on the screen, with some of the 384 characters not used. In the second pass, any remaining characters are added to the character allocation of C and the screen image is completed. If buffer C should run out of data, the remaining lines are added to D and displayed.

Summarizing the screen loading policy, two passes are made over the four data buffers:

1) Pass one allocates L/32 * 1024 characters to each buffer where L is the desired number of lines.

A. If any buffer runs out of characters, allocation at line request is not filled on the first pass.

B. If a buffer runs out of data, the remaining lines of the line allocation (from the allocation register) are added to D.

C. If the line request is filled, no further processing is done on that buffer.

2) Pass two allocates all the remaining characters to the buffers, as needed up to 1024 when all processing stops. D is processed last.

A. If any buffer runs out of characters, then there are no characters left for any other buffer. The screen will display all that was added on the first pass, plus whatever was added up until the 1024 character limit was reached.

B. If any buffer runs out of data, then the remaining lines of its allocation are added to D. No character allocation is altered since on the second pass characters can be used as needed until 1024.

C. If any buffer completes its line allocation, processing on it stops. Again the total number of characters remaining are simply available to the other buffers.

The preceding screen loading policy was somewhat simplified since all additions to the screen image made during the first two pass processing are made only in integral lines, i.e., half of a line of text is never displayed on the screen.

6. OPERATION AND FORMATS

The VCP is a program which is loaded into a PDP-8 equipped with 8K of core memory, a 32K disk, and a 637 interface. Assuming a compatible dataphone system properly connected to both the 637 interface and the Sanders 720, the program as it presently exists will accept information from the teletype as the main information source. It will then allow the user to execute "address specifications", "rolls", "allocates", and "switches", all of which have been previously discussed. The system presently requires all rolls--either up or down--to have the number of lines to be rolled specified by the user in the command to the system. The command is typed in the spaces directly following the FBLK> which is always present on the screen. The formats are:

ROLLUP A 5

ROLLDN A 3

ROLLUP B 7

ROLLDN D 3

ALOCAT N, M, P, Q

SWITCH

Letters typed on the screen after the FBLK>symbol not in the above list will be assumed to be a command to the main information source and will be typed out on the teletype under the present system. As mentioned previously, address specification from the console is indirect. Specifically, the system assumes that the main information source will transmit as its first six bits of data, an A, B, C, or D as specified by the user. If no specification is made, the first six bits can be anything. The information typed on the teletype to simulate a main information source must be formatted as follows:

First character-buffer-A,B,C or D

Second character - switch or allocation -S (switch)

or ! (allocation)

If an allocation is to be transmitted, the next four characters are assumed to be the binary number equivalent to the number of lines of each buffer desired.

The next characters may be anything except the @ or > . These codes are illegal because the @ is given at the termination of a message; the > is used, when transmission to the user is desired, to complete the message and initiate the screen-loading algorithm. Provision is made for filling any number of buffers at once. If another buffer is to be filled before anything is displayed on the screen, the above sequence can be repeated.

The VCP, as it exists, will handle any number of control units, memories, or screens simultaneously. The number of screens (in ASCII code) is loaded into location 0-0020, the number of memory units into 0-0021, and the number of control units into 0-0022. Buffer areas are automatically set up on the disk as required. Buffers not used occupy no room on the disk.