

Rec

SOURCE: DCE 26 NOV 63

TITLE: Preliminary thoughts on using our computer facilities to help us in our computer programming activities.

COPIES TO: JHW, DCL, LJC, CPB, TLH, FKT, BJH, SDM, EBS

SEE ALSO ITEMS: DCE 25 NOV 63, LJC 23 NOV 63

FILE FOLDER: Research Planning

FORMAL DESCRIPTORS:

ABSTRACT: Real-time feedback during writing or debugging a program, and the management of a ~~program-dis~~ program-design records (instruction ~~listings~~ listings, flow-diagram-like listings, etc.) are cited as the two most likely special aspects of our computer-aided working system to concentrate upon in the next few months, and some general discussion of the ~~latter~~ latter is given.

TEXT: In thinking about the "coordinated working system" that we seek to evolve for our own use (see DCE 25 NOV 63), and wishing also to get representative aspects of this working system under way relatively soon to form the basis of a demonstration of our accomplishments for the year, I have generated the following thoughts to add to the base of material for us to ~~discuss~~ discuss.

1. A "coordinated working system" is the focus for planning.
2. Miscellaneous facility for handling documentation files can be shown, but in a relatively short time, we couldn't make too much headway on an integrated working system covering all of our file systems within the allotted time--that is, headway to have unusual facility in using ~~te~~ them.
3. The area where it would seem we could make headway, in an integrated sense, would be in the programming design and checkout area of activity. For instance, suppose we concentrated upon the following:
 - a. ~~Real-time~~ Real-time feedback in checking out and debugging. In FKT 12 NOV 63 there are a ~~number~~ number of suggestions, but only three of the nine seem likely within a relatively few months: Items 5,6 and 9, i.e. stopping operations at any point with flexible hard-copy printouts available and easy way of restarting at same point, an accounting on request of machine times with additional feature of ~~fe~~ being able to set time limits on runs, and a flexible choice of real-time traces with ^{xw} automatic ~~e~~ dumping and restarting. Perhaps other possibilities can be assembled and evaluated. Will need to assess the potential state of our basic system facility for the months ahead in order to determine which ~~real-te~~ real-time feedback processes are feasible to consider implementing and integrating into the working system.

Constructing and using process-structuring records in the design, debugging, ~~modification~~ ~~modification~~ and documentation (for others) of computer programs. Without too much new programming required, but with establishment of conventions, terminology and procedures, we should be able to do some very useful and interesting things.

4. Discussion of some process-structure design-recording possibilities:

a. John Wensley has described to me in the past some notions about the tabulations and ~~labelling~~ ~~labelling~~ labelling that might be appended (automatically or semi-automatically) to the listings of instructions (listed in symbolic form). I don't remember the details, but I ~~we~~ ~~believe~~ believe that he mostly was considering at the time the possibilities within the scope of aids ~~that~~ ~~that~~ that a human clerk could provide--but some of the aids could be made automatic.

b. In this level of work, with the instructions that are fed to a compiler or assembler, the record established within computer store for a real-time computer aided interrogation, or established on hard copy for off-line use, would seem to offer relatively fertile ground for new conventions, terminology, procedure. For instance, listing of the "jumped-from places" (from what other points in the ~~po~~ ~~po~~ program could control have jumped to this point?) could be worth while.

c. A next level up in the records for a program's design is what usually is ~~satisfis~~ ~~satisfis~~ satisfied with a flow diagram. Len's recent memo (LJC 23 NOV 63) indicates a start toward a recording structure not dependent upon graphics. I should think that we can pursue this approach very usefully over the immediate months ahead. For instance:

(1) study ~~ch~~ ~~ch~~ categorization of ~~sem~~ ~~sem~~ segments (what is usually described in a process box), there may be more, useful ~~lable~~ ~~lable~~ labels than usually indicated by the graphical symbology in flow charts. Labels can easily be attached, in the text-character symbology we consider using, to fit many categories, sub-categories, etc.

(2) Not only categories of "nodes" of our processing network, but the "links" as well might be categorized usefully. i.e. besides categorizing and labelling the statements (what used to be in flow-chart boxes), we consider categorizing and labelling the types of links existing between different statements (or what used to be the lines between the flow-chart boxes).

(3) Find a convenient way to hop between the segment-level description (what used to be portrayed in simple flow charts) and the instruction-level description (the ~~lis~~ ~~lis~~ listings of instructions). For example, perhaps the description of what a segment is supposed to do no longer is coupled as a comment that is part of the listing, but now is removed to a linked position in the linked-segment graph (I guess I am searching for good names while I write. "Flow chart" doesn't quite seem appropriate. It really is a matter of a description of segments of programs and of their linkage. Linked-segment record?). You should be able to jump from a view of one form to a view of the other with relative ease.

(4) There are probably often good reasons for having higher-level flow charts in complex programs. I know that this is the case for complex equipments and ~~thiZiW~~ their ~~blockZiW~~ block diagrams. A set of conventions for establishing and portraying these different levels of process organization would be useful. Probably the set would be the ~~seZiW~~ same no matter at what level you would be working, once you had gotten above the first level or so.

(5) At any level, it should be easy to follow paths along the various branches and linkages. On-line procedures ~~waZiW~~ would do well to harness light-pen selection of branches to follow. Off-line would require some indexing as to where in the hard copy would be found the location to which this branch takes you next. Indexing in a common index is one possibility, but it ~~soZiW~~ would be more convenient to have, for instance, the page number associated with a given branching link listed right at the branch indication.

(6) If primitive line-drawing technique becomes available in our display (or in the hard-copy output), we may find it very useful as one type of record to portray just the network topology for the linked-segment record at a given level. Perhaps nodes would be labelled by some brief ~~codeZiW~~ code--point with light pen to get a description of that node displayed for you. The storage-tube display might come in handy for such a network display, since establishing vectors by dot generation can be done economically if they don't have to be regenerated continually.

(7) Changes in the record at one level usually imply changes at the other levels. Potential automatic accounting done--if you have made a change at one level, the other-level locations where change is potentially implied could be tagged appropriately, or a central file of up-dating changes (potential) could be kept. These records of potential changes to be made could link back to the explicit change which caused them to be established, so that ~~weZiW~~ when you began ~~you~~ ^{your} cleanup work you would know where to begin.

(8) Provision for hanging miscellaneous comments onto the recorded structure should be made. By both the author and others. Along with this could go a central file of "annunciators" or flags that you (the author) could ~~reviseZiW~~ review occasionally to see what ~~comments,ZiW~~ comments, suggestions, warnings, etc., could have been left by others who may have been studying or using the records or the actual program.

(9) The off-line techniques being developed ^{now} for composing and modifying text could undoubtedly be ~~expended~~ ^{expanded} to include the composing and modification of these various types of process-structure design records. Same goes for the simple on-line review and editing possibilities with ~~hiZiW~~ the 160-A-only system we are developing. Also, the techniques for off-line designation of changes to prior records, and for the extraction, modification, and integration into new hard-copy records.

26 NOV 63

5. I realize that the development of programs for all of these capabilities is beyond the scope of the next few months of activity, but the planning for coding, terminology, structuring conventions (format) etc., would want to ~~now~~ be made with considerations of the future trends of our system, and I think that consideration of these future possibilities is important to us now.