- - - - - - - - - - - - -
D R A F T   O N L Y
- - - - - - - - - - - - -

8 August 1989

Linked Overlap Management for System Design and Organization Development

The author is with the test unit in Quality Assurance for Branch of LIS, Bureau of Land Management, Department of the Interior, Denver Service Center.

NOTICE: This paper was not produced at government expense.

ABSTRACT:

A new method for the development and management of computer system software is proposed. It is based on applying the complexity metric of McCabe to the linking pin function of Pelz and Likert and in the context of DOD-STD-2167A. The method is extended to organizational development and is named linked overlap management (LOM). It is shown to be a universal management model which is appropriate for projects in government.

KEYWORDS:

complexity metric of McCabe, DOD-STD-2167A, linked overlap management, linking pin function of Pelz and Likert, organization development, software development life-cycle, universal management system

INTRODUCTION:

"In government, it is easier to obtain forgiveness than to get permission." -- paraphrased from Admiral Grace Hopper

This introduction describes the complexity metric of McCabe (CMM), the linking pin function (LPF) of Pelz, and DOD-STD-2167A (STD) of the U S Department of Defense. The discussion section describes the application of the methods.

CMM applies to data flow graphs. CMM also applies to organization models where logic flow is expressed. A standard model for organization management in business is LPF. A standard model for organization development in government is STD. This paper applies CMM to reduce LPF into a less complex model. The simplified LPF model is then applied to the organization method of STD. The new method is named linked overlap management (LOM), and it is generalized as a universal management model.

CMM [McCabe 1976] is described as the number of discrete data sets required to completely exercise the paths of a flow model. For example, consider the flow graph of Figure 1.

\- - - - - - - - - - - - -

Figure 1 about here

\- - - - - - - - - - - - -

The number of regions (r) is 3, the number of edges (e) is 9, and the number of nodes is 7. CMM is expressed as:

$$\text{regions} + 1 = 3 + 1 = 4, \text{ or}$$
$$\text{edges} - \text{nodes} + 2 = 9 - 7 + 2 = 4.$$

This means that four data sets are needed to completely test each path in the flow graph.

LPF [Likert 1961] is described as a theory of overlapping groups within an organization. The linking pin is one who is a member of two such groups, as a manager and as a subordinate. For example, consider the organization of Figure 2 [Young 1979].

\- - - - - - - - - - - - -

Figure 2 about here

\- - - - - - - - - - - - -

The linking pins are represented by "0" and others by "O". As a linking pin manager, one's authority depends on how much the subordinates allow to be exerted over them. Hence authority flows upward in the organization. Authority is also determined by how much one is influenced by subordinates in that one's authority exerted upward and laterally is determined by one's authority exerted downward.

Supportive relationships also affect LPF [Lippitt 1967], not as observable phenomena but rather as a condition of personal worth. The management style which best effects LPF is the Participative-Group system [Likert 1967] from which came System 4 for reducing conflict [Likert, Likert 1976]. The relative success of LPF can also be determined by human asset accounting which measures return-on-investment of employees [Likert 1967]. However the current value of employees is determined by productive capacity and customer good-will. Hence if able personnel leave then the value of the organization is reduced.

STD [DOD-STD-2167A] is described as system design development which consists of a logical flow of computer software configuration items [CSCI]. These are divided into both computer software components [CSC] and computer software units [CSU]. For example, consider Figure 3 [DOD-STD-2167A].

- - - - - - - - - - - - -

Figure 3 about here

- - - - - - - - - - - - -

One CSCI may not contain other CSCIs. One CSC may contain other CSCs and CSUs. One CSU may not contain other CSUs. STD contains the necessary requirements for quality assurance, traceability, and accountability. However the management of STD at times may seem puzzling due to its comprehensiveness and to the sparse implementation commentaries in the literature. This difficulty is addressed in the discussion below.

DISCUSSION:

Experience has shown that CMM < 10 is desirable for computer program modules written in high level languages such as FORTRAN [McCabe 1983]. This principle applies to simple flow graphs such as trees. A binary tree is a series of nodes which have only two-way branching. For example, consider Figure 4 which has two balanced nodes as leaf nodes and a CMM of four.

\- - - - - - - - - - - - -

Figure 4 about here

\- - - - - - - - - - - - -

As another example, consider Figure 5 which has six balanced nodes as leaf nodes and a CMM of six.

\- - - - - - - - - - - - -

Figure 5 about here

\- - - - - - - - - - - - -

Balanced binary trees of higher orders clearly produce CMMs > 9. Thus the most complex balanced binary tree with CMM < 10 is that of Figure 5.

From Figure 2 the simplest link pin unit is in Figure 6.

\- - - - - - - - - - - - -

Figure 6 about here

\- - - - - - - - - - - - -

However this is not a balanced binary tree. The desirability of a balanced binary tree is illustrated in the basic rule of construction for a standard outline. An item A may have parts numbered 1 to N. However if an item A has only one part then it is not numbered 1 but rather incorporated back into item A as not really being a part. Because the binary trees here are ultimately related to management theory below, the same line of reasoning applies. If a link pin manager has only one subordinate, then the span of control of the manager is too shallow and the manager and subordinate should both be moved laterally or upward. Therefore to simplify the link pin unit [LPU] into a balanced binary tree, the number of subordinates must be changed from three to two. Figure 7 shows the conversion which has a CMM of six.

- - - - - - - - - - - - -

Figure 7 about here

- - - - - - - - - - - - -

The following discussion is based therefore on a CMM of six.

The STD design of Figure 3 is also converted to the LPF model of a balanced binary tree in Figure 8.

- - - - - - - - - - - - -

Figure 8 about here

- - - - - - - - - - - - -

The simplest LPU of Figure 8 with a CMM of six is in Figure 9.

- - - - - - - - - - - - -

Figure 9 about here

- - - - - - - - - - - - -

It is this LPU which forms the basic building block for the system design and organization development below.

CONCLUSION:

Applying the LPU to the STD may be traced from the Software Design Document [SDD] through the coding phase of the software development cycle. The preliminary and detailed design SDDs contain LPUs to describe all CSCIs. The coding phase systematically address the design structure of the SDDs. However the coding phase could implement the SDDs differently but in such a way as not to violate the SDDs. For example, consider this case statement structure in Ada:

```
CASE variable IS
                  WHEN '01' => csci_a ... ;
                  WHEN '02' => csci_b ... ;
                  :
                  WHEN '26' => csci_z ... ;
END CASE ;
```

LPUs of coding groups work on the component CSCIs of the SDDs such as A through Z here. But the implementation of the CSCIs into the opening case statement above does not reflect the SDD design per se. Hence it is possible when implementing LPUs to the STD to have a design and implementation of different structure but fully compliant and compatible to STD.

This seeming difference in design and implementation is managed easily however by applying the structure of LPUs at the SDD levels to management with an identical design. This is named linked overlap management (LOM). Its the highly structured organization should succeed as well at management as has its design implementation in STD.

ACKNOWLEDGMENTS:

Thanks are due to John Deuter, Mary Gaylord, and Judith Harrington for helpful discussions.

REFERENCES:

DOD-STD-2167A

Military Standard. Defense System Software Development. Washington, D C: Department of Defense. 1988.

Likert 1961

Likert, R. New Patterns of Management. New York: McGraw-Hill. 1961.

Likert 1967

Likert, R. The Human Organization. New York: McGraw-Hill. 1967.

Likert 1976

Likert, R; Likert, J G. New Ways of Managing Conflict. New York: McGraw-Hill. 1976.

Lippitt 1967

Lippitt, G L. Organization Renewal. New York: Appleton-Century-Crofts. 1967.

McCabe 1976

McCabe, T J. "A Complexity Metric." IEEE Transaction on Software Engineering. 1976. SE-2. 101-11.

McCabe 1983

McCabe, T J. Structured Testing. New York: IEEE Computer Society Press. 1983. Catalog No EHO 200-6.

Young 1979

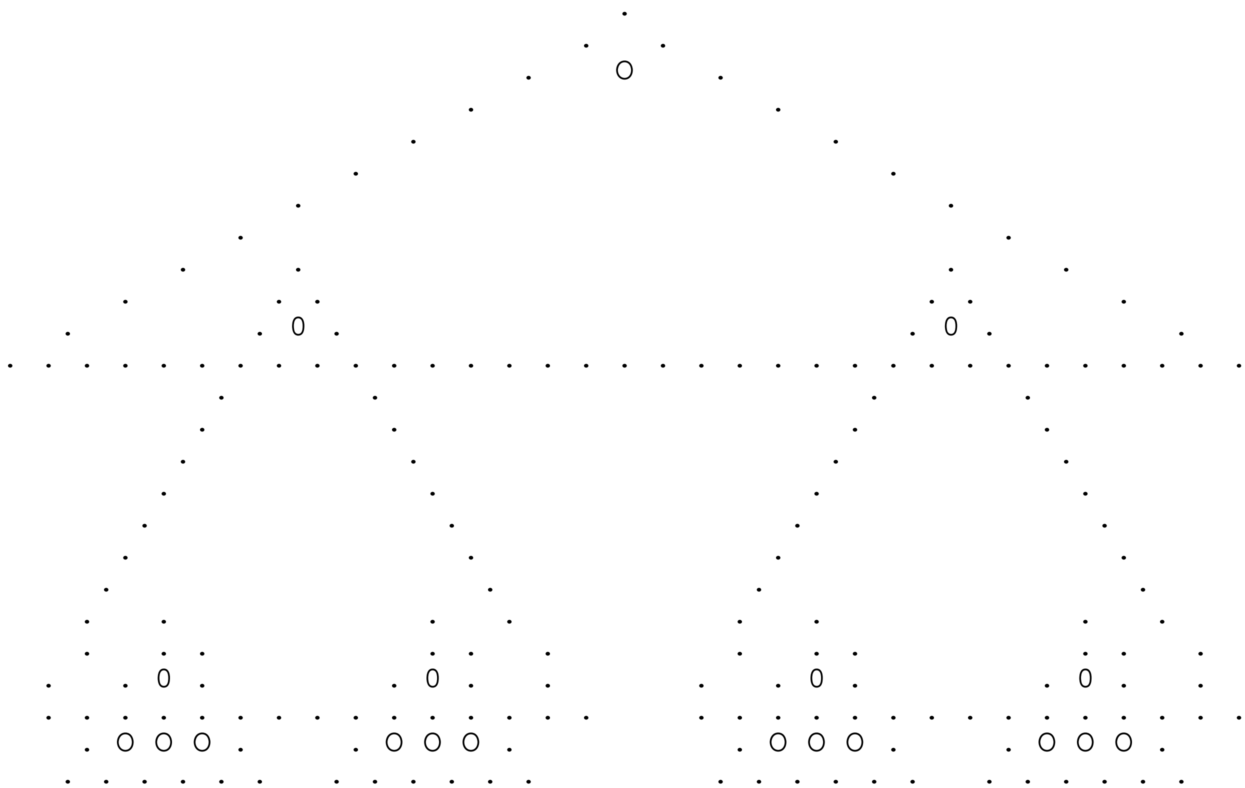
Young, J N (compiler). Management: Theories and Practices. Washington, D C: National Defense University. 1979.

[figures follow]

Figure 1

```
             Node
              :
              : (e)
              v
        ...> Node ....
        :     :      :
    (e) : [r] : (e)  : (e)
        :     v      :
        .... Node    :
                     :
                     :
    ........ Node <...
    :         :
    :   [r]   : (e)
    :         v
    :   .... Node ....
    :   :            :
(e) :   : (e)   [r]  : (e)
    :   :            :
    :   ...> Node <...
    :         :
    :         : (e)
    :         v
    .......> Node
```

Figure 2



```
Figure 3
--------
                          ............
                          :          :
                          :   CSCI   :
                          :..........:
                               :
                      .........:..........
                      :                  :
                ......:..              ..:......
                :       :              :       :
                :  CSC  :              :  CSC  :
                :.......:              :.......:
                      :                  :
           ...........:..                :
           :            :                :
     ......:..    ......:..          ....:....
     :       :    :       :          :       :
     :  CSU  :    :  CSC  :          :  CSU  :
     :.......:    :.......:          :.......:
                        :
              ..........:..
              :           :
          ....:....   ....:....
          :       :   :       :
          :  CSU  :   :  CSU  :
          :.......:   :.......:
```

Figure 4

```
           ............
           :          :
           :   NODE   :
           :..........:
                :
       .........:..........
   (e) :                  : (e)
.......:..              ..:.......
:        :              :        :
:  NODE  :      r       :  NODE  :
:........:              :........:
       :                  :
   (e) :..................: (e)
                :
           .....:......
           :          :
           :   NODE   :
           :..........:
```

Figure 5

```
                          ............
                          :          :
                          :   NODE   :
                          :..........:
                               :
                      .........:..........
                  (e) :                  : (e)
               .......:..              ..:.......
               :        :              :        :
               :  NODE  :              :  NODE  :
               :........:              :........:
                      :                  :
           ...........:..              ..:...........
       (e) :            : (e)      (e) :            : (e)
    .......:..   .......:..          ..:.......   ..:.......
    :        :   :        :          :        :   :        :
    :  NODE  : r :  NODE  :    r     :  NODE  : r :  NODE  :
    :........:   :........:          :........:   :........:
           :            :              :            :
       (e) :............: (e)      (e) :............: (e)
                      :                  :
                  (e) :..................: (e)
                          .....:......
                          :          :
                          :   NODE   :
                          :..........:
```

Figure 6

```
               .
              . .
             . 0 .
            . . . .
           .       .
          .         .
         .           .
        .             .
       .               .
      .                 .
     .                   .
    .   .             .   .
    .   . .           . .   .
  .   . 0 .         . 0 .   .
  . . . . . . . . . . . . . . .
    . O O O .     . O O O .
   . . . . . .   . . . . . .
```

Figure 7

```
               0
              . .
            .     .
           0       0
         . .       . .
        .  .       .  .
       O   O       O   O
```

Figure 8

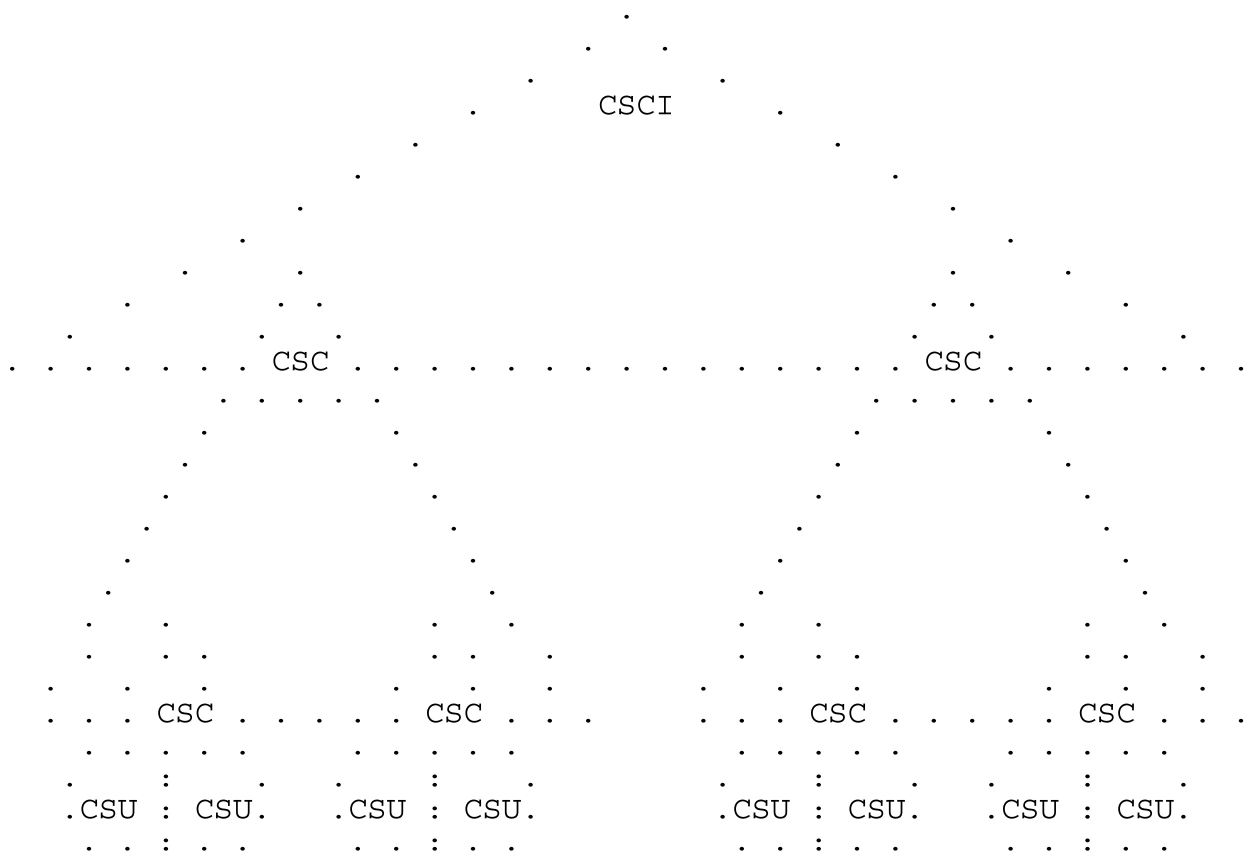


Figure 9

```
       .       .
      .       . .
     .       .   .
    .       . CSC .
   . . . . . . . . . . . . .
          .         .
         .           .
        .             .
       .               .
      .                 .
     .                   .
    .   .             .   .
    .   . .           . .   .
  .   .   .         .   .   .
  . . . CSC . . . . . CSC . . .

    . . . . .     . . . . .
   .    :    .   .    :    .
   .CSU : CSU.   .CSU : CSU.
    . . : . .     . . : . .
```