February 25, 1976

26 FEB 76 9: 02

W. D. Royall

Did you see this? Please return.

B. Grad

We had not, thanks for calling it to our attention.

Bill

3/1

2/26/76

Burt,

We had not--thanks for calling it to our attention.

W. D. Royall

Aleksa per our conversation. BG 3/1. Call to

March 2, 1976

Mr. S. A. A[illegible]ka MAR 3 9 15 AM '76

Per our conversation.
Please return.

B. Grad

DELTA TERMINALS AND LIGHT PENS MAKE YOUR SELECTION EASIER

We make it easier to select the right terminal, and easier to select the right information on the terminal's display.

We've always made it easier for you to match your video display to your video display problem, by offering terminal features such as:

- 7x9 dot matrix for the most readable upper/lower case characters in the industry
- Custom keyboards designed for your special needs
- 3072 character memory
- 27 line by 80 character display
- Exclusive PAGING capability that permits recovery of top and bottom line rolloffs

These and many other unique and useful features in DELTA 5000 video display terminals work to save time, money and effort, add convenience, minimize error possibilities and eliminate unnecessary computer time.

We want to make one more point about simplifying your selection. Our new light pen takes the time and tedium out of selecting required information on the display screen. The light pen is easy to use, accurate, and it can save considerable time. All DELTA 5000 terminals are available with a light pen as an option.

Now that we've told you how easy it is to simplify your terminal selection, we'd like to show you with a demonstration at your convenience. Contact us today to set one up, or for literature or applications assistance.

Delta Data Systems Corporation

Woodhaven Industrial Park
Cornwells Heights, PA 19020
(215) 639-9400

DELTA DATA SYSTEMS, LTD.
London: 01-580-7621

Service in 42 locations in the U.S. and 13 European countries and Canada.

CIRCLE 84 ON READER CARD

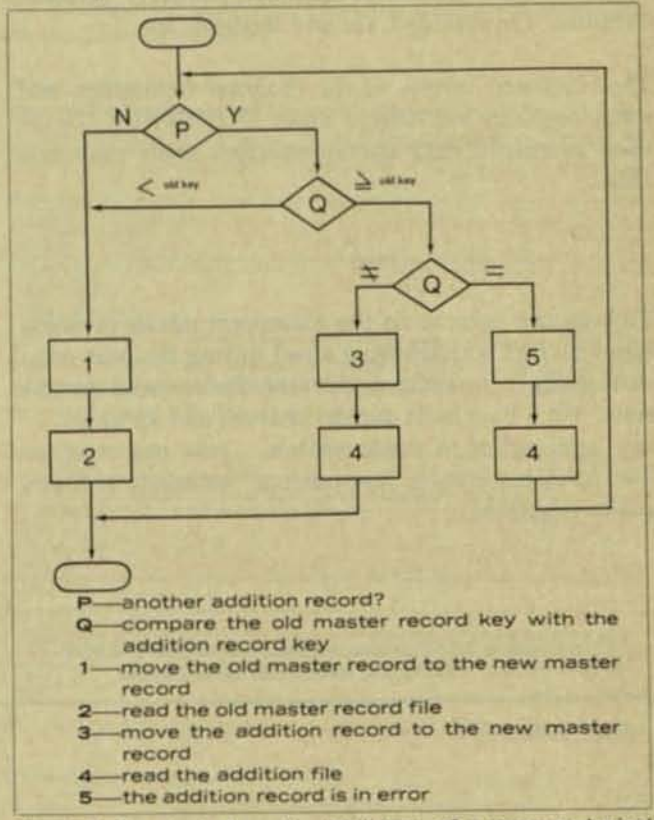
the forum

The Dark Side of Structured Programming

Many positive things have been said about structured programming. Maybe the time has come to be somewhat negative.

The proximate cause of this article was a reading of IBM's Independent Study Program on Structured Programming [ed. note: A review appeared in the October issue, p. 27]. In it we learn that the flowcharting decision box is now a predicate node; the connector is a collector node; and the process box, a process node. We also learn that to convert our spaghetti-bowl flowchart thinking to structured thinking, we must master the techniques of interchange, transposition, combination, resolution, and substitution. With help like this, we may have nothing to worry about—structured programming may not survive.

Perhaps we should begin again. Let's try it with a not-so-



P—another addition record?
Q—compare the old master record key with the addition record key
1—move the old master record to the new master record
2—read the old master record file
3—move the addition record to the new master record
4—read the addition file
5—the addition record is in error

Fig. 1. A flowchart done the "old" way, of a common logical situation: merging additions to a master file.

	1	2	3	4
P ?	Y	Y	Y	N
Q (LESS THAN CASE) ?	Y	N	N	—
Q (EQUAL CASE) ?	—	Y	N	—
1	X	—	—	X
2	X	—	—	X
3	—	—	X	—
5	—	X	—	—
4	—	X	X	—

Fig. 2. The same situation as Fig. 1 presented in a decision table.

new remark: From a general point of view, structured programming isn't a new thing. It's just another way of saying that you should organize and localize your code. However, don't think that by this remark, I mean to run down the structured programming concept. I don't. It shows how to organize and localize code in a methodical way, and that's to the good, since it allows us to minimize art and maximize procedure.

However, it's hard to overcome a strong sense of déjà vu, and with good reason, because we decision table devotees have been here before. The fact is that if you use decision tables to organize your program logic and adhere to one simple rule, you've organized your logic according to structured programming principles. The simple little rule is: No GO TO's in the action stub. Actually, there's one important exception to this rule, but rather than bother you with it here, I've buried it at the end.

Now, how do decision table and structured programming logic differ? The answer is shown in a nutshell in Fig. 1, which represents a common logical situation. The key at the bottom of Fig. 1 substitutes values for variables to transform the general case shown into a familiar specific one—the merging of additions into a master file.

Decision tables have no trouble with the logic shown. (See Fig. 2.) However, this common logic situation is the nemesis of structured programming—to express this logic in structured form requires the introduction of a flag (F), as shown in Fig. 3. So, yes, decision tables and structured programming do, on some occasions, produce different logical structures for the same situation, but in those in-

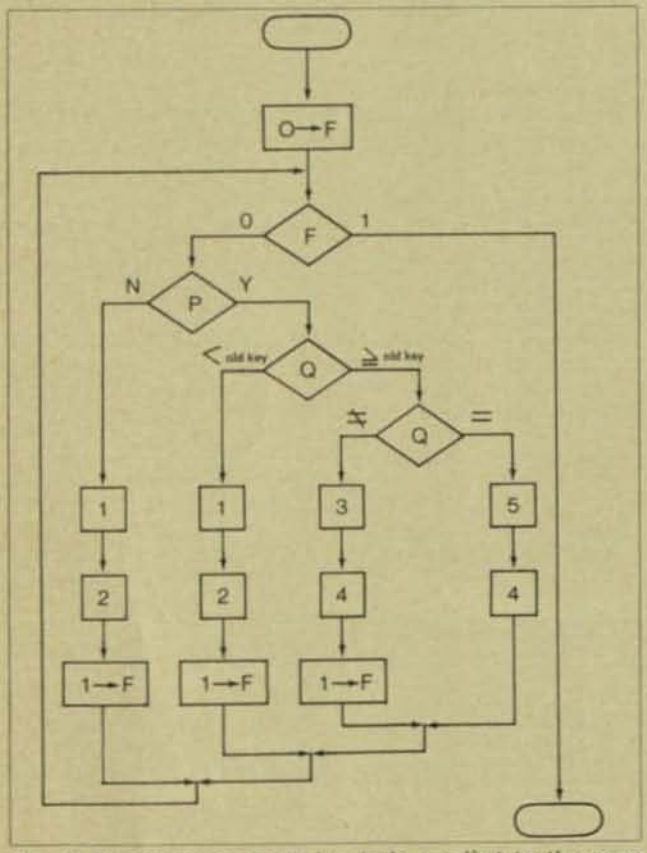


Fig. 3. Structured programming logic, applied to the same situation as Figs. 1 and 2, requires flags and increases in complexity.

stances where they differ, who needs the contortions of structured programming?

It could be contended that logic such as that shown in Fig. 1 should be allowed, and that otherwise, structured programming principles are to be followed. But now the door is ajar and who knows what other spaghetti bowls may creep in? Decision tables aren't hampered with the need for

Shrink

THE ULTIMATE DATA COMPRESSION FOR ALL 360/370 USERS

While everyone else is ordering more disk drives you can turn some of yours back! AMAZED? It's easy with SHRINK, a general purpose superpowerful data compression package that uses a combination of many different techniques to achieve phenomenal data space reduction.

SHRINK's compression rates average from 20 to 85%. Here are some sample benchmark results of SHRINK's compression.

Previously compacted files with much binary data	20-30%
OS object decks	43%
Report type data files	70-85%
File data consisting of alphameric numeric and some binary items	60-80%

Compression/expansion is highly efficient requiring an average of only 1 CPU second per 100K bytes on a S/370/158. Sophisticated redundancy checking assures data integrity.

CPU overhead is minimized in random keyed retrievals and in transaction/master mtaching as only selected records need expansion. Further, compressed files can be sorted directly, reducing sort times and sort work space. SHRINK will speed up I/O bound jobs by permitting more records per block size and fewer volumes per file. SHRINK is easy to use from any batch or on-line program. Installation and training takes an hour. A no obligation trial is available.

- Ability to sort, retrieve or update compressed files without expanding them,
- Improved sort times and reduced sort work space requirements,
- Record compression/exapansion subroutines callable in the standard manner from BAL, COBOL, FORTRAN, PL/I, etc.,
- Stand-alone file compression/expansion utilities,
- A guarantee that the expanded record is an exact replica of the original, before compression,
- Ability to be easily implemented in an on-line monitor environment,
- Reduction of turnaround time for I/O bound jobs resulting in more throughput in the computer as well as better job performance, and
- Reduction of tapes and disks, tape drives and disk spindles.

PROGRAMMING METHODS
Proprietary Software Division
1301 Avenue of the Americas
New York, N.Y. 10019

☐ Send me details about SHRINK

NAME_____
Company_____
Address_____
_____
Phone_____ CPU_____

CIRCLE 95 ON READER CARD

PROGRAMMERS AND ANALYSTS

Free Employment Service

Serving Northeast, Southeast and Midwest U.S.

- Scientific and commercial applications
- Software development and systems programming
- Telecommunications
- Control systems
- Computer engineering
- Computer marketing and support

Call or send resume or rough notes of objectives, salary, location restrictions, education and experience (including computers, models, operating systems and languages) to either one of our locations. Or check the reader service card for a free sample resume. We will answer all correspondence from U.S. citizens and permanent residents and will guarantee our best efforts in a professional and ethical manner to all qualified applicants that we think we can help. Our client companies pay all of our fees.

RSVP

RSVP SERVICES, Dept. M
Suite 700, One Cherry Hill Mall
Cherry Hill, New Jersey 08034
(609) 667-4488

RSVP SERVICES, Dept. M
Suite 300, Dublin Hall
1777 Walton Road
Blue Bell, Penna. 19422
(215) 629-0595

RSVP SERVICES

Employment Agency for Computer Professionals

CIRCLE 121 ON READER CARD

EDP SEARCH

WHY IS FOX-MORRIS THE FIRST CHOICE OF EDP PROFESSIONALS?

A recognized staff of professional experts offering LOCAL, REGIONAL & NATIONAL COVERAGE through our direct branches as well as coast to coast coverage through our 75 member firms of National Personnel Consultants, Inc.

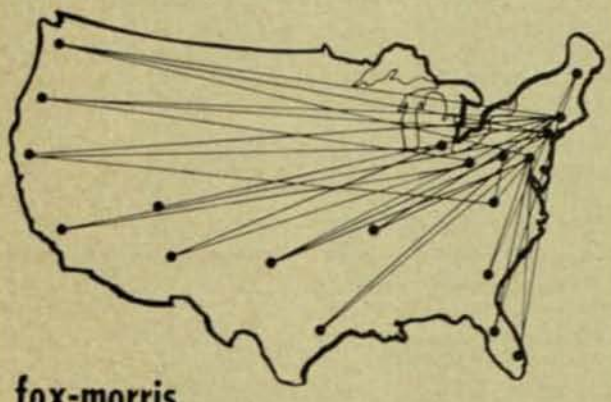
Completely confidential service geared to your career objectives.

Unparalleled contacts with industry based on years of successful results.

Professional resume evaluation and career guidance.

Client companies assume all employment costs—including interview expense, relocation (if necessary) and search fee.

SEND RESUME DIRECT, OR CIRCLE NUMBER BELOW ON READER CARD.



fox-morris personnel consultants

Philadelphia: (215) 561-6300, 1500 Chestnut St., Philadelphia, Pa. 19102
New York: (212) 697-1820, 633 Third Ave., New York, N.Y. 10017
Pittsburgh: (412) 232-0410, 6 Gateway Center, Pittsburgh, Pa. 15222
Wilmington: (302) 654-4465, 2005 Concord Pike, P.O. Box 7017, Del. 19803
Princeton: (609) 452-8135, P.O. Box 2063, U.S. Rt. 1, Princeton, N.J. 08540
Baltimore: (301) 296-4500, 409 Washington Ave., Baltimore, Md. 21204
Charlotte: (704) 527-4980, 4000 Park Road, Charlotte, N.C. 28209
Atlanta: (404) 321-3888, 2200 Century Pkwy., N.E., Atlanta, Ga. 30345

CIRCLE 116 ON READER CARD

the forum

any such compromises.

Decision tables have another advantage over structured programming. When you're finished with your decision tables, you can just turn them over to a preprocessor, and out comes your code. With a structured flowcart, you must still write all that COBOL code. And if you thought writing COBOL code made your hand hurt, wait until you write COBOL code in structured format. All those duplicate operations create a real temptation to slip in a few nice, simple GO TO's to avoid writer's cramp.

Preprocessed spaghetti

Now, it's possible to contend that while decision tables organize logic well, preprocessors turn out spaghetti bowls. True enough, but this is the same as saying that even if you avoid GO TO's in your COBOL code, the compiler is going to generate a lot of branches, which is true. The point is that you maintain programs at the level at which you write them, and with the use of preprocessors, you maintain your programs at the decision table level.

There's also the fact that with decision tables, there are automatic procedures which can be built into the preprocessor to:

1. guarantee that all possible combinations of conditions have been considered
2. eliminate all redundancy
3. detect all contradictions.

No flowcharting technique, including structured flowcharts, can claim to do all three.

That one exception

Here's the buried exception part mentioned earlier. Consider the situation in Fig. 4. Now, suppose that condition B depends on the outcome of action D. The situation then has

TABLE—X	1	2	3	4
CONDITION A	Y	Y	N	N
CONDITION B	Y	N	Y	N
ACTION C	X	X	—	—
ACTION D	X	X	X	X
GO AGAIN	X	—	X	—
EXIT	—	X	—	X

Fig. 4. The one case where the simple decision table like this presents difficulties without a GO TO is when condition B depends on the outcome of action D.

TABLE—X	1	2
CONDITION A	Y	N
ACTION C	X	—
ACTION D	X	X
PERFORM TABLE—Y	X	X
EXIT	X	X

TABLE—Y	1	2
CONDITION B	Y	N
GO TO TABLE—X	X	—
EXIT	—	X

Fig. 5. The situation of Fig. 4 with a GO TO in the action stub.

to be described as shown in Fig. 5. This is the only type of situation in which a GO TO action should appear in the action stub.

. . . once again

Going back to the starting point, if you use decision tables, you don't need to know anything about predicate nodes, collector nodes, process nodes, interchange, transposition, combination, resolution, and substitution. That alone should be enough to sell you on decision tables.

—Thomas R. Gildersleeve

Previously with Univac for 14 years, for the past six years Mr. Gildersleeve has been a consultant specializing in organization studies, and in developing project management procedures and system development standards.