Behavior Research Methods, Instruments, & Computers
1984, 16 (6), 545-547

Conjunction of a programming language and text formatter for generating randomized questionnaires

ANNABEL J. COHEN and JOAN E. FOLEY
University of Toronto, Scarborough, Ontario, Canada

The conjunction of a progamming language and a text formatter is described as an aid to the construction of questionnaires in which the order of presentation of items is randomized and the output has a neat and professional appearance. The technique has marked advantages over the use of a programming language on its own. Modifications can be easily and independently made in the format of the document or in the experimental procedure itself, for example, number or type of questionnaire items, or instructions. The particular example is for BASIC and RUNOFF, but the technique would generalize to other language-formatter pairs available on computers of any size.

Many psychological research problems employ questionnaires. Typically, a large number of subjects are asked to complete the same questionnaire. In order to avoid effects of the particular order of presentation of items upon the response, different forms of the questionnaire are generated. In the ideal case, each subject receives a different random order of the items. Typing each questionnaire individually requires inordinate effort for the return. A word processor and text formatter will reduce some of the work but will not select items in a random order. A programming language, on the other hand, will generate random orders but will not control the appearance of the output with the flexibility of the formatting software. The researcher who is familiar with the advantages of software for both formatting and programming therefore faces a dilemma. If the programming language is chosen for its ability to randomize items, the appearance of the questionnaire may be sacrificed. If the formatter is chosen for generation of a few preselected random orders, the advantage of complete randomization is lost. This paper focuses on the possibility of having the advantages of both the programming language and the text formatter.

In hindsight, the use of a programming language to generate text-formatter commands for production of individual randomized questionnaires has marked advantages over the use of either software on its own. Yet, upon embarking on the programming problem of questionnaire development, these advantages may not be obvious. Furthermore, the possible disadvantages of working simultaneously within the constraints of both a text formatter, for example, RUNOFF, and a programming language, for example, BASIC, might discourage the attempt to integrate the two systems. Finally, there may be a dissuading fear that incompatibilities between the two systems would prohibit their joint application. Our experience, however, has been positive. We therefore draw attention to the technique.

A questionnaire resulting from the integration of RUNOFF and BASIC is well formatted. Legibility is therefore enhanced, which has implications for consequent attention given to the task by the subject. For example, scale items are centered regardless of the word length of the scale dimension names: for example, with the dimension name placed below the numerical scale

low 1 2 3 4 5 high
confidence

or with the dimension indicated at each pole

very unconfident 1 2 3 4 5 very confident

If, during the course of the studies, it becomes necessary to adjust page size, a change in one command is all that is required in order to retain proper formatting. The same accomplishment without a text formatter would require either the alteration of many PRINT statements or else the inclusion, within the BASIC program, of text-formatting routines that already exist within RUNOFF. Similarly, the decision to single, double, or triple space can be executed with one command. Underlining is also possible. Pages are numbered automatically.

Because few commands are required to produce a well-formatted document, unfamiliarity with the text editor should not present a major obstacle. For example, estab-

The work was supported by grants from the Natural Sciences and Engineering Research Council of Canada made to J. E. Foley, Department of Psychology, Scarborough Campus, University of Toronto. The commentary on earlier drafts by Howard L. Kaplan of the Addiction Research Foundation, Toronto, is gratefully acknowledged. The first author's mailing address is now: Centre for Research in Human Development, Erindale Campus, University of Toronto, Mississauga, Ontario, Canada L5L 1C6.

lishment of the overall structure requires the following three commands:

```
.sd 80 [specifies line length and general conventions]
.sp 2  [specifies double (2) spacing]
.TP 3  [specifies beginning of a new page when (3)
       blank lines remain]
```

In addition, other useful commands are:

```
.c;    [to center]
.pg    [start a new page]
.nf    [to print as indicated without "filling"]
.f     [to "fill" the space for justification]
.b     [skip a line, start a new line]
.i5    [indent (5) spaces]
^& /&  [placed around text for underlining]
```

In BASIC, a randomized order for the questionnaire items is generated for each subject, and rules for assigning condition names for subjects in particular groups are established. For example, if there are N subjects and N/2 females and N/2 males, and if s is the subject number:

```
 90  p$="Female"
100  for s=1 to N
110  if (s=(N/2)+1) then p$="Male"
120  print #1 ".c;"p$;" Subject";s; " Experiment 1"
```

In this example, the sex of subject and subject number, along with other title information, would be printed on the questionnaire, centered by the ".c;" command to RUNOFF. Thus,

Female Subject 1 Experiment 1

is produced. Similarly, other variables in the instructions can be entered into the text.

At least two separate files are generated by the BASIC program. The first is the text file that will be sent to the RUNOFF program for text formatting. RUNOFF, in turn, uses that text file to produce a ".mem" file that, when printed, becomes the formatted questionnaire. The second file produced by BASIC contains only the information about the experimental conditions and randomized order of trial types for each subject. For example, File 2 contains the information that Subject 1 in Condition 1 is female and is to receive, on the first trial, Trial Type 7, and on the second trial, Trial Type 3, etc. Such a file might look like:

```
F 1
1 7
2 3
. . .
```

Because it includes the information about trial number and trial type for each subject, it serves a second purpose as an integral part of the subsequent data-entry process. Upon completion of data collection, a data-entry program written in BASIC will take this file as input and prompt the user at the keyboard for the subject's response for each trial. In the present specific example, suppose that the female subject in Condition 1 responded "100" on the first trial (Trial Type 7). The respective line generated on the computer screen by the data entry program would be

Trial 1 Type 7 - - - ?

and 100 would be keyed in at the terminal. Subsequently, the prompt for the response for Trial 2 Type 3 would follow. The restatement of the trial-type information ensures that data are correctly entered for subsequent analysis. To facilitate data entry, the trial-type information can be printed unobtrusively on the questionnaire in brackets next to the trial number if one wishes, although it is of course not necessary.

Theoretically, there is no limit to the number of files that can be generated by the questionnaire program. Probably only one of these will be for formatted text, whereas others will be used for data management and analysis.

The use of RUNOFF actually produces very few additional constraints on writing the program. Those familiar with RUNOFF, however, will be disappointed by the constraints of BASIC upon text entry. Whereas most lines typed into an editor in preparation for RUNOFF contain fewer than three characters of overhead in addition to the text itself, a line of BASIC that produces a line of RUNOFF input must include a line number, print command, file number, quotation marks and punctuation to separate variables from the literal text. These problems are characteristic of BASIC regardless of the use of RUNOFF, and, indeed, the use of RUNOFF certainly simplifies the number of BASIC statements required to produce well-formatted text. At the same time, the use of BASIC reduces enormously the number of statements to be keyed as the input file to RUNOFF. For example, each question need not be printed in full in the program when the same structure of the question repeats throughout. Moreover, repeated aspects of the questionnaire, for example, the use of a rating scale, can be referred to in a subroutine that is called in the program when required. Finally, the same program serves for all subjects.

The program can be modified for related research by editing in BASIC or with a text editor. The time spent in program development, therefore, readily promotes more than one particular experiment.

Our use depended upon the simultaneous access to BASIC and RUNOFF on one computer, in this case a DECsystem 1090 with the TOPS-10 operating system. With this configuration, only six brief commands or replies to prompts by the user at the keyboard were required after entry into BASIC in order to produce the final printed output.

The technique of integrating a programming language with a text formatter, however, is not limited to BASIC and RUNOFF. For example, PL/1 (language) and

SCRIPT (formatter) are a compatible pair for IBM mainframes. Generally, both a suitable programming language and text formatter are available on computer systems of all sizes, from microprocessors to mainframes. However, for the technique to be productive, the programming language should have good alphanumeric capabilities; FORTRAN IV, for example, is not a good choice, but FORTRAN 77 is adequate.

Sample programs and output are available from the authors.

(Manuscript accepted for publication December 7, 1984.)