A USEFUL METHOD OF COMMAND CONSTRUCTION IN FORTRAN SOFTWARE SYSTEM

Jojima, Kuniyuki
Department of Mathematics, Kumamoto Women's University

Tanaka, Kiyoshi
Department of Information Systems, Interdisciplinary Graduate School of Engineering Sciences, Kyushu University

Asano, Chooichiro
Research Institute of Fundamental Information Scinece, Kyushu University

https://doi.org/10.5109/13337

出版情報：Bulletin of informatics and cybernetics. 20 (1/2), pp.71-77, 1982-03. Research Association of Statistical Sciences

A USEFUL METHOD OF COMMAND CONSTRUCTION IN FORTRAN SOFTWARE SYSTEM

By

Kuniyuki JOJIMA*, Kiyoshi TANAKA**

and

Chooiichiro ASANO,*****

(Received November 26, 1981)

Abstract

In order to realize new commands of data-management and data-analysis in a software system, already constructed by a FORTRAN language, a useful method is proposed using the new syntax and the format-table, which is available through TSS terminals.

1. Summary and Introduction

In view of the compatibility and independence of software system for various computer systems and models, the software system is usually written by a standard FORTRAN language, including the monitor part of the system. On the other hand, individual FORTRAN programs are developed on the basis of theoretical studies for mathematical analysis.

In these situations, in order to realize such new mathematical methods in the software system, already constructed or been constructing by the standard FORTRAN language, an easy method is needed to implement new commands to the software system by the use of individual FORTRAN programs of mathematical analysis.

In the present paper, such an easy method is proposed with a syntax format-table studied through TSS terminals. The proposed method is actually applied to NISAN (a New Interactive Statistical ANalysis) system and the illustration is demonstrated.

2. Principle of Command Registration

Most of data-analytical software systems are planned to have various commands of data-handling and mathematical analyses, and are designed to process effectively and sometimes interactively data-files. Therefore, it may be possible to say that commands involved present main functions of the software system for the users.

* Department of Mathematics, Kumamoto Women's University

** Department of Information Systems, Interdisciplinary Graduate School of Engineering Sciences, Kyushu University 39

*** Research Institute of Fundamental Information Science, Kyushu University 33

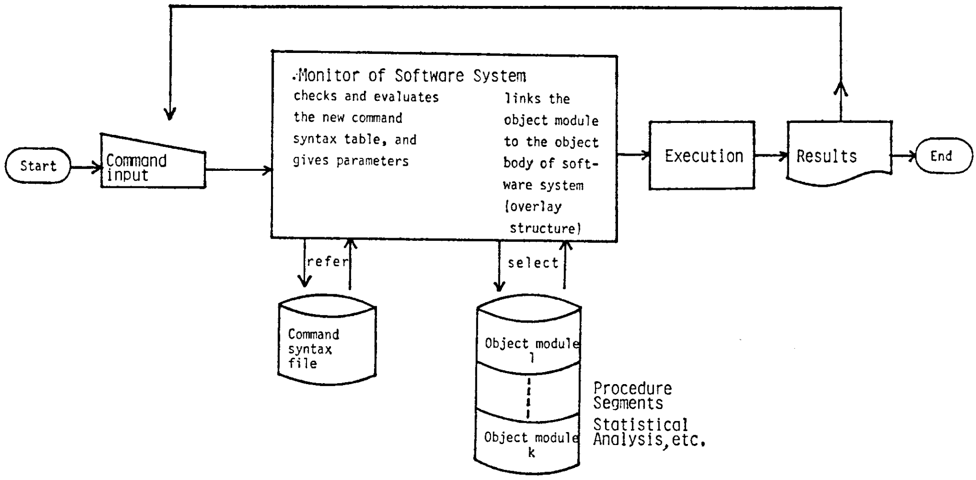

Fig. 1. Flow to get a result

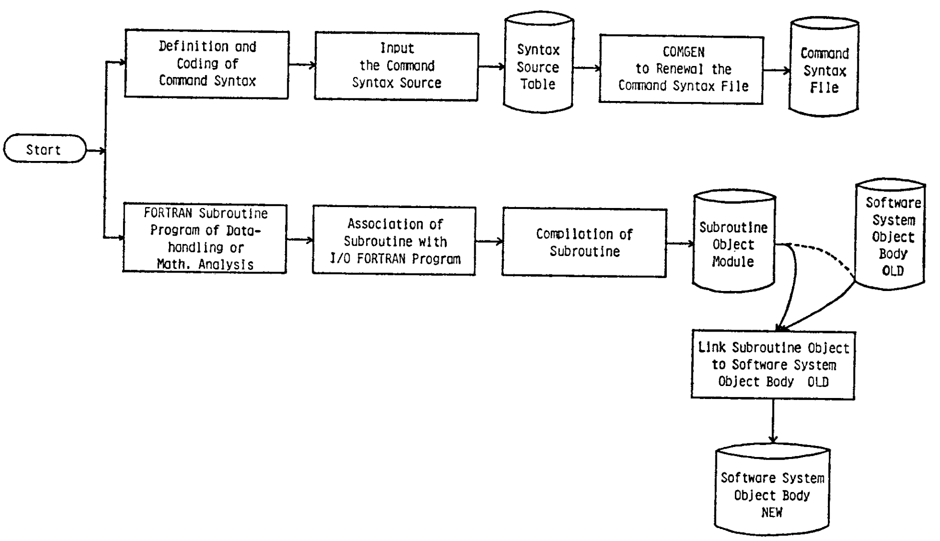

Fig. 2. Block diagram of command registration

Regarding to these commands, the functions of the monitor of software system have very important role to interpret input-commands, to allocate memories, to lead to files, and to control several peripheral devices, terminals and programs. Figure 1 shows an illustrative flow to get a result, starting from an input-command, as a bird-view of monitor in NISAN software system.

Therefore, the command processing by the monitor of software system is firstly to refer the command syntax to a file of syntax table already defined, as the checking for input. Secondly the monitor gives parameters needed for processing, and delivers to the concerned segments, i.e. individual programs of actual treatments, with an overlay structure. When the software system, including the monitor, is written by a standard FORTRAN language, the convertibility of the software system for other computer systems is so easy that the method of supplement of new commands must be required. Especially when the user is a researcher on data analysis in his proper field, it may be frequently natural for him to add several new commands with his originality to the software system, converted to his computer system. Thus the necessity arises for the addition of new commands from such researchers in the users of software system.

The proposed procedure of the method of supplement composes of three steps as follows,

(i) addition of the syntax table, to which the monitor refers,

(ii) formation of a FORTRAN program for a data-processing, and segmentation,

(iii) linking the FORTRAN program with the monitor of the software system.

These parts are mutually independent, but no contradiction must be allowed. In Figure 2, a block diagram is given as the proposed procedure mentioned above.

3. Definition and Coding Form of Command Syntax

The definition of command syntax given here is very simple and is shown as

COMMAND OPERANDS,

where some keywords, specified arbitrarily by the user, may be included in the operands. For instance, having a function of the Hotelling's T^2 test for one sample theory, i.e. a test of a sample mean vector drawn from a multivariate normal distribution with unknown variance-covariance matrix, a command MEAN1 may be defined with several operands by

MEAN1 [FI (file name)]* [VAR (variable name list)]* M̲EAN (mean vector specified) [A̲LPHA (significance level)],

where [ ] means that operand and item involved can be omitted, [ ]* means that operand and item involved are not necessary when they are fixed previously by LOCK command, { } means that only one operand must be selected from the plural operands shown, and the underlines of operand mean the shortening form. In order to implement the new command into the syntax source file, a supporting program COMGEN (COMmand GENerating), written by FORTRAN, is applied. That is to say, a table of newly defined command syntax is at first made in the working area on the basis of the syntax rule of symbolic language in COMGEN. To perform easily this work, a useful form of the syntax coding sheet is prepared, and is shown in Figure 3. Then the COMGEN is applied to the table in the working area and the syntax source file is renewed. Thus the registration of the new command is completed. A series of such operations is available on TSS mode.

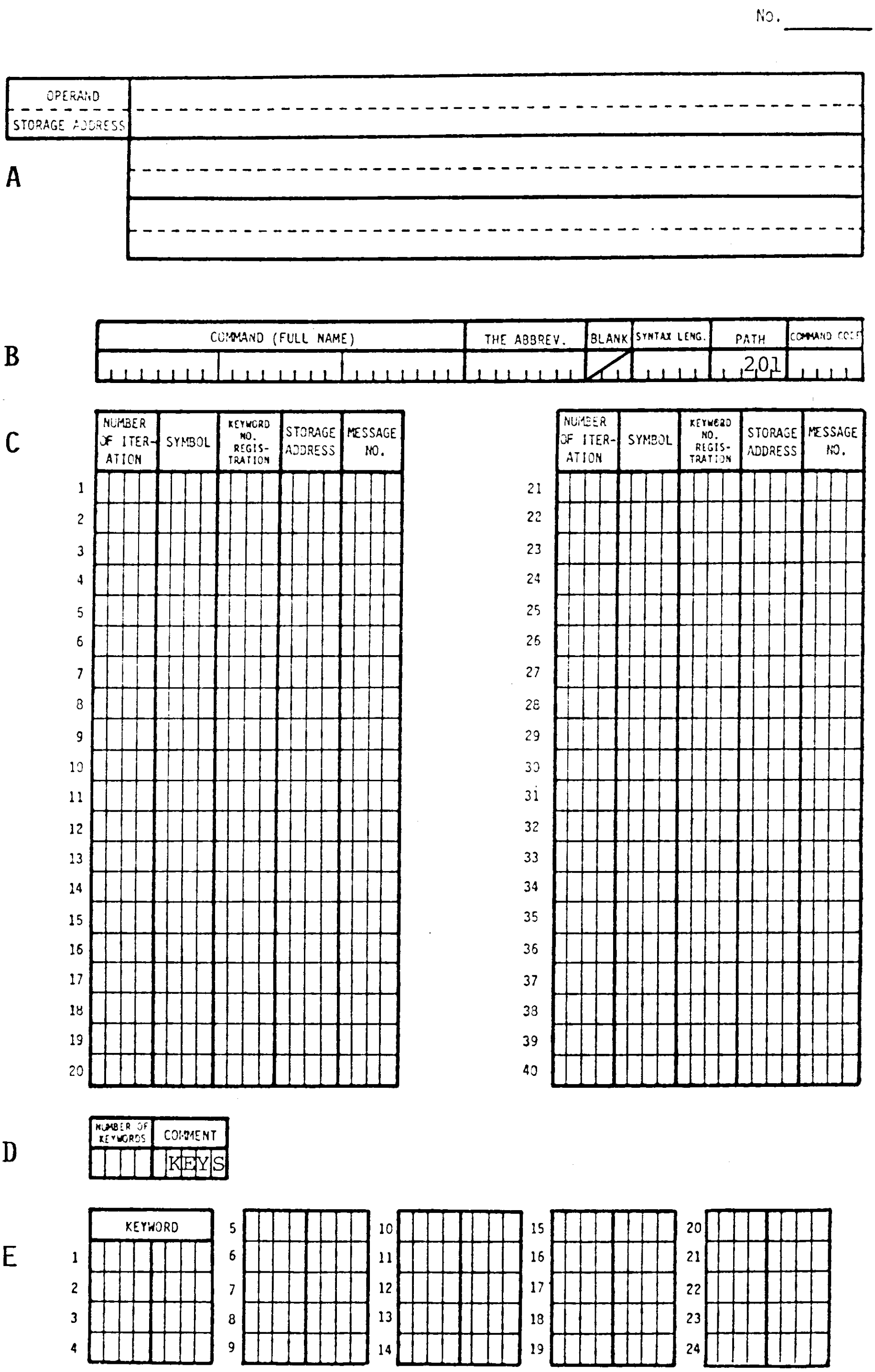

Fig. 3. **COMGEN** syntax coding sheet

4. Command Syntax Rule in COMGEN

In order to arrange a newly defined command syntax on the basis of the rule of COMGEN, a convenient syntax coding form is prepared, which consists of five parts, i.e. from Part A to Part E. Every row in the respective parts is uniformly regarded as an IBM card with eighty columns.

To fill in the respective rows in the sheet, one has to obey the command syntax rule of COMGEN and to use several symbols with specified semantics. The details in each part are given in the following way.

At Part A, the operands of command and the addresses to store are assigned. At Part B, The command name is assigned closely to the left side, and the length of syntax is given after filling in the steps (rows) at Part C. Part C is a main part of formation of the command syntax, and several symbols are used with the respective semantics. The number of iteration at the first column is entried closely to the right side, when only the symbols *(and & are used. Varieties of symbols entried at the second column are defined in Table 1. Tte keyword No. or the registration flag at the third column is entried closely to the right side, when the symbols shown in Table 2 are used.

Table 1. Symbols used in the second column at Part C

	Symbol	Syntax
(a) Item	I	Integer
	F	Real number
	V	Vector variable
	A	Alphabetical data
(b) Keyword	&	Keyword
(c) Control character	/	Starting symbol of syntax
	≪ ≫	Omission symbol, where [ ] is used in operand
	< + >	Option symbol, where { | } is used in operand
	/* FREE */	Symbol to show arbitrary order of input items, where /* Item1 FREE Item2 FREE Item3 */ is used in operand
	*(	Starting symbol for iteration
	)	Closing symbol for maximal number of iteration. When $N*(A)$ is used in operand, this means that item A is input N times at the maximum
	)¥	Closing iteration by keyword
	)≠	Closing iteration by iteration number in the same command
	);	Closing iteration by completion of input
	):	Closing iteration by completion of keyword
	END	Ending syntax, where syntax starts from / and completes at END

Table 2. Symbols used in the third column at Part C

Symbol	Syntax
V	Setting 1 without any condition.
&	Entrying the respective keyword No. for keywords in Part E.
)¥	Entrying the respective keyword No. for keywords in Part E. Then the iteration is closed by the keyword. Whenever 99 is entried, the iteration is closed by any keyword.
)#	Entrying the same as the storage address of item with iteration.
):	The same for)¥.

The storage address at the forth column is entried closely to the right side, where the storage address of the starting symbol / is entered the maximal value of the storage addresses. The storage addresses of input and output files of the software system may be fixed conventionally to be 1 and 2, respectively. The message No. at the fifth column is entried closely to the right side, when symbols / and &, or items I, F, V, A are entried. The message No. of / is the index of information for the HELP command, and No.'s of the other symbols are indices of the error messages for their inputs. At Part D, the number of keywords used in the command is entried closely to the right side. The maximal number of keywords is limitted to twenty four. At Part E, keywords of eight characters at most are entried closely to the right side.

5. An Illustration

To illustrate the useful method proposed in the present paper, let us show the procedure to implement a new command MEAN1, which is already shown as an example in section 3. For this purpose, it is enough for us to show merely how to define and apply the syntax coding sheet and how to use the command syntax rule in section 4 since the COMGEN program operates after the completion of inputting the coding sheet.

Now the command MEAN1 is planned with several operands as follows,

MEAN1 [FI(A)]*[VAR(50*(V)¥)]*MEAN(50*(R)¥)[ALPHA(F)]

where the number of variables is designed to be not greater than fifty. Then the whole of preparation are to fill in the coding sheet before the operation of COMGEN, and is shown in Figure 4.

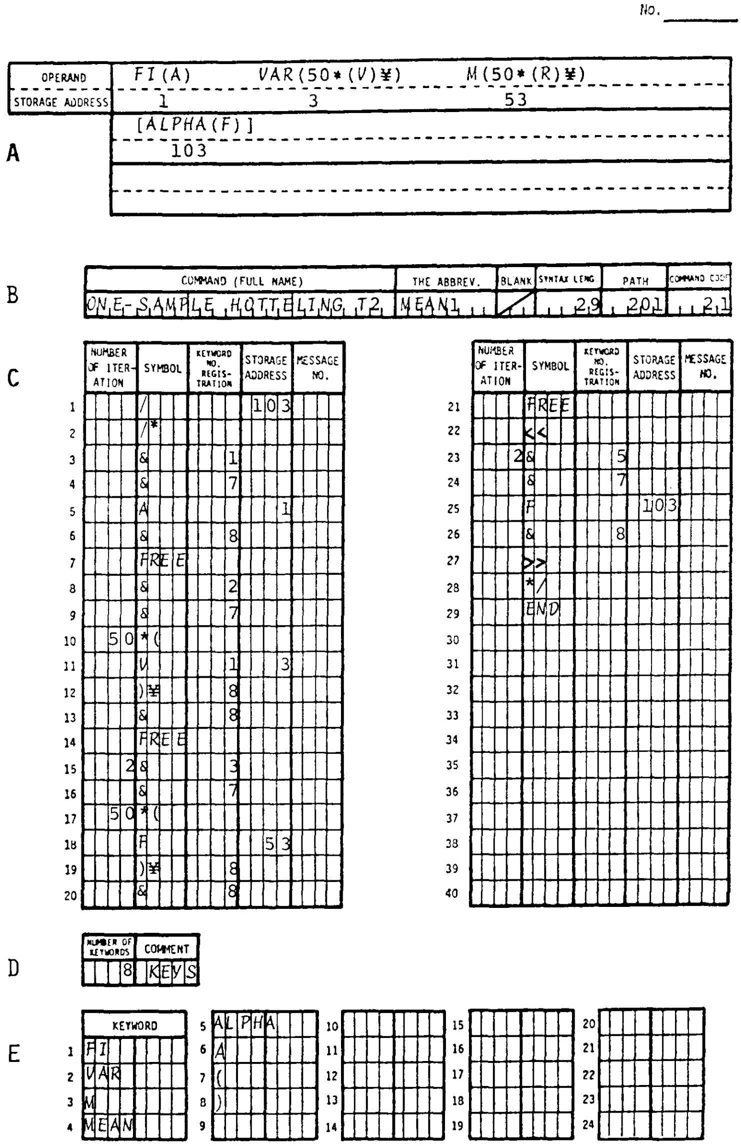

Fig. 4. An illustration in case of MEAN1

References

[1] Asano, Ch., Wakimoto, K., Shohoji, T., Komasawa, K., Jojima, K., Goto, M., Tanaka, Y., Tarumi, T. and Ohsumi, N.: *The statistical principle and methodology in NISAN system*, 3rd Proc. Comp. Statist. (1979), 431-437.

[2] Asano, Ch., Watanabe, M., Hosokawa, K. and Kurihara, K.: *A recent development in NISAN system*, Bull. Inform. Cyber., 1, (1981).

[3] Asano, Ch., Tanaka, K., Jojima, K., Nishigori, N.: *Command Construction Manual for NISAN Research Members*, Res. Rep., No. 101, Res. Instit. Fund. Inform. Sc., Kyushu Univ., (1981).

[4] Asano, Ch., Watanabe, M., Hosokawa, K., Kurihara, K. and Tanaka, K.: *NISAN System—Users' Guide*, Res. Rep., Res. Instit. Fund. Inform. Sc., in press.

[5] Fujitsu Co., FACOM OS IV/F4 TSS Comment, Fujitsu Manual (1976), (in Japanese).