A Study on Information Systems

(A paper indited by a group of research workers from the Central Statistical Board—Romania)

Preliminary studies have shown that the automation of the statistical data processing requires by conventional means an excessive span of time. Some works aimed at increasing the analysis programming productivity were carried out by the Computing Centre of the Central Statistical Board. This paper deals with a system of programs particularly meant for the automation of analysis of current statistics processing.

1. BRIEF DESCRIPTION OF THE S.I.S. (STATISTIC INFORMATION SYSTEM) METHOD

The SIS method is based on the premise that the analyzing and programming of the applications of economic data processing requires a certain automation (it follows that even an untrained personnel would be able to perform the conventional analyzing and programming). For this purpose, a system of definitions regarding the information system is being worked out. These definitions are later used in the analysis of information systems (the systematic description of the information system structure, including the checking of dynamic efficiency requirements being satisfied), and in the information system synthesis (the design of the software from the output specifications, with optionally given inputs according to the specifications of a given equipment).

The SIS system will be set up as consisting of three basic sections:
SIS 1 — for dynamic analysis of the technological processes;
SIS 2 — for analysis of the information system;
SIS 3 — for the information system synthesis.

2.1. General definitions

Analyzed economic unit: field confined to a closed surface, information entering and coming in and out of it.

Information system, restricted meaning: the set of data included by the economic unit under analysis, including data coming in and out of the surface limiting the analyzed economic unit.

Information system, comprehensive meaning: the information set, including the equipments used for information converting and/or processing and/or transmitting.

2.2. Postulates regarding the information systems, restricted meaning

1° Within an information system, the set of information I is partitionable by denomination. The set of information having the same name make up a major file f:

$$f \in I \tag{1}$$

2° A major file is partitionable, at least according to the nature of some information sub-sets, formed of discrete elements described as identification sub-sets.

A sub-set S of a major file, of the same nature as the identification sub-set and of the same value of at least one element of the identification sub-set is called *a file:*

$$S \in f \tag{2}$$

3° A file is divisible by the values of all the elements of the recorded identification sub-sets:

$$i \in S \tag{3}$$

4° There are certain converting operations of some files into others, by means of some operators A called algorithms:

$$S_e := A^k(S_i) \tag{4}$$

where

S_e = the set of resulted files;

S_i = the set of given files;

A^k = the processing algorithm.

Introducing the time variable (t) and taking into account the physical restrictions, we can write relation (4) as follows:

$$S_e(t) \cong A^k S_i(t - T_p) \tag{4'}$$

where

T_p = processing time (the duration between the time when the processing of the first record in the set of input files starts and the time of output of the last record in the output file, admitting that both sets have a finite number of records; if not so, it is admitted that a sufficient number of input records are extracted and the processing time is so determined).

5° The processing operations are performed inside certain domains space-limited by closed surfaces and called processing points P:

$$S_e : = \frac{P}{A}(S_i) \tag{5}$$

Corollary: The equivalent diagram of the information circuit within an information system includes files of information and information processing points; consequently, it can be represented as a graph.

2.3. Definitions regarding the information system, restricted meaning

Information input medium: The physical medium that stores the records of data files for running and/or keeping and/or processing.

Example: punched cards, according to a model regarding a certain type of recording.

Processing flow chart: conventional representation of the graph corresponding to an information system, the titles of the implied information files and the names of the algorithms.

Example: suggested in figure 1.

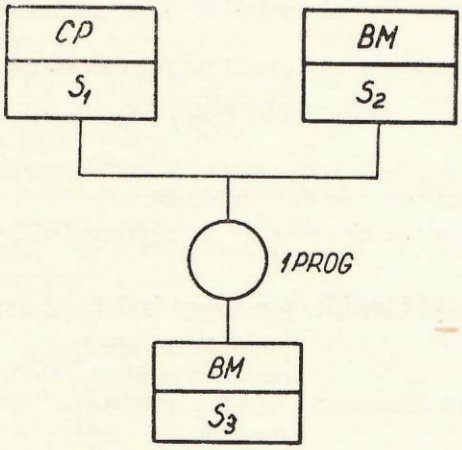


Fig. 1

Type of input medium:
CP — punched card; *BM* — magnetic type.
File name:
S_1, S_2 — input files; S_3 — output file;
1*PROG* — correcting algorithm.

Cyclical information system: an information system within which the information file processing is periodical in time.

The inquiring information system: information system, characterized by the fact that the data file circulation and processing is nonperiodical in time (on request).

Input file: Information file that enters the considered processing point (from the inside or the outside of the information system).

Output file: information file coming out of the considered processing point outside the information system, as defined in paragraph 3.1., directed to another processing point (possibly inside the information system).

Examples:

— The print-outs produced by a computer, for the processing point represented by the management of the analyzed unit represent an output file;

— the data punched cards that are destroyed after their reading in by the computer (they are of no interest after that) are *not* an output file although information comes out the processing point.

Structure of the standard information file (Fig. 2). A standard information file comprises 3 sections:

— A record including at least the data required for establishing the file identity (for example file name or code), as well as any other data required for the information file processing. This record is called "file identification record" and is noted by *IS*.

— Any number of records characteristic or that file (noted r_i; i — order number of the record).

— A record specifying the file end: as this may be standardized (for ex.: **** or 99), no special notations are necessary, because its existence is implied.

It follows that a standard information file can be represented as:

$$s = IS(r_i) \tag{6}$$

(the brackets do not separate here a variable and a function, but suggest that the *IS* information is a common denominator).

The file identification record may be represented by an order number, possibly implied.

The structure of the identification record is defined as:

— file name	(possibly implied)
— file code	(possibly implied)
— data of file working-out	(possibly implied)
— file source	(possibly implied)
— file purpose	(possibly implied)
— code of input medium [1])	(possibly implied)
— version number [1])	(possibly implied)
— periodicity	(possibly implied)
— locking key [2])	(possibly optional)
— period of file retaining	(possibly optional)
— copy number	(possibly optional)
— comments	(possibly optional)

[1]) Definition follows

[2]) Secret coded information, in order to prevent access of undesired users to the information comprised in the file.

Periodicity: the period by which the considered data file or parts of it enter or come out the analyzed unit or the considered processing point.

The structure of the specific record

By definition, the record is made up of [fields] (the most simple, indivisible pieces of information, making up a record). The set of fields of a record may be divided into 2 sub-sets:

— record identification field, containing the information required for identifying the record in the file; by definition, the identification fields comprise qualitative information, i.e. the contents of these fields can always be represented by a finite number of whole numbers (by an adequately defined coding);

— data fields containing qualitative and/or quantitative information associated to the record (these may be possibly implied; for example, in the case of an enumeration of individuals classified by different criteria, the record regarding an individual may be formed of the identification fields only, specifying the classes the individual belongs to , that is the quantitative information, +1 being implied).

In accordance with all this, it can be writen:

$$r_i = IR_i(D_i) \tag{7}$$

where

IR — record identification fields;
D — data fields.

The structure of record identification fields. By definition, the set of the identification fields has the following structure:

— record name (optional);
— record number (optional);
— a number of fields containing qualitative information (codes);
— a number of associated data fields (optional);
— fields specifying the maximum length of the record;
— any comments (optional).

The data fields structure. By definition, the data fields set has the following structure:

— data name (ND) (optional)
— file input time (optional)
— fixed file output time (optional)
— number of data fields (optional)
— a number of data fields, composed of:

a) a number (figure), representing the quantitative information associated to the record;

b) the phase associated to the quantitative information (the phase in relation to the file periodicity)

The length of a record is defined as the number of characters contained in a record. The records may be of two types:

a) records of a fixed length — the structure regarding the contents is independent, so that the length is invariable (the missing quantitative data are entered as null values);

b) records of a variable length — the structure is determined by the contents; in this case, the length of the record is variable (the number of record fields increases as the information volume increases).

The noting convention of a data file. In accordance with the above definitions, a condensed notation is:

$$s = IS(IR_i(D_i(P,\ F_i)) \tag{8}$$

where

P = the file period (conventionally written as associated to each data field);

F_i = the phase associated to a data field.

An equivalent notation points out the data field:

$$D_i^k = D_{ij}^{k\ IS_k}{}_{IR_{ij}}\ (P^k,\ F_{ij}) \tag{8'}$$

where

k = file code

i = record/file code

j = data field/record code

That is to say, the data files are formed, by definition, of indexed time functions.

2.4. Elements of data file algebra

Identity of 2 files. Two files are identical if all the records are equal with each other [1]) and occupy the same position in the file. In this case, one file may be called the *copy* of the other:

$$D^{1}_{ij\ IR_{ij}}(P^1,\ F_{ij}) \equiv D^{2}_{ij\ IR_{ij}}(P^1,\ F_{ij}) \tag{9}$$

IS_1 $\qquad$ IS_2

$$i = 1 \ldots n$$

$$j = 1 \ldots m$$

The equivalence of 2 files. Two files are equivalent if all the records are equal, but have not the same position in the file.

Example: A sorted file is equivalent with the unsorted initial one. The equivalence operation is marked as $\sim$.

The equality of 2 files: Two files are equal if, by common factoring some fields, two identical files are obtained.

---

[1]) The contents of all the fields are equal.

Example:

$$IS(IR_1, D_1, IR_1, D_2, IR_1 D_3) = IS(IR_1 (D_1, D_2, D_3) = IS\ IR_1 (D_1, D_2, D_3) \quad (10)$$

NOTES

a) In the example above, the field identification information was considered as implied.

b) It is obvious that, by common factoring, the minimum file is the file where all the possible field common factoring operations have been performed.

The logical equality (of type 1) of two files. Two files are logically equal (of type 1) if:

— data identification information,
— data names,
— data fields number and structure,
— data periods and phases,
— file identification information (except the version name), are equal.

It is said that one file is a version of the other.

The operation is noted by ^{o}.

Example: An updated stock file in relation to the original stock file.

The logical equality (of type 2) of two files. Two files are type 2 logically equal if the logical equality conditions are satisfied, except the equality of the phases.

Memory files. If, with reference to a processing point, an input file and an output file can be defined as equal or logically equal (of type 1 or 2), then it is said that there exists a memory file (memory) bearing the name of the two files.

NOTE

Had not the concept of output file destination been specified when defining the output files, then any input file would have become a memory, except the case when the information processing would have been destructive.

Cyclical processing. The processing operation, periodically performed in order to obtain output files from input files, according to a given processing program (a sequence of primary instructions, corresponding to the data processing technique, defining precisely the processing operation). By definition, when the processing is initiaded the required input information (program file and input files) is available.

NOTE

By definition, the period of the program file is longer than the processing period.

Access to data file. At the processing point, the processing successively produces records of the output files from records of the input files. It follows that a storing zone must exist in the processing point.

The diagram of a processing point is given in figure 3.

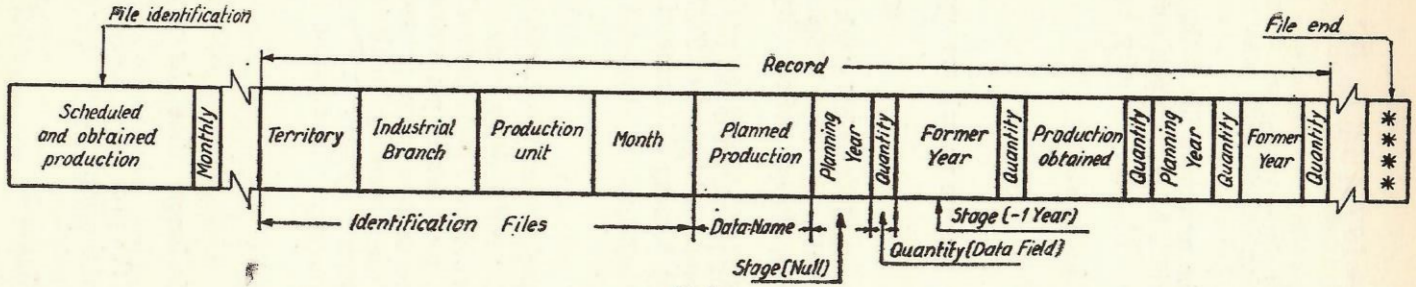


Fig. 2

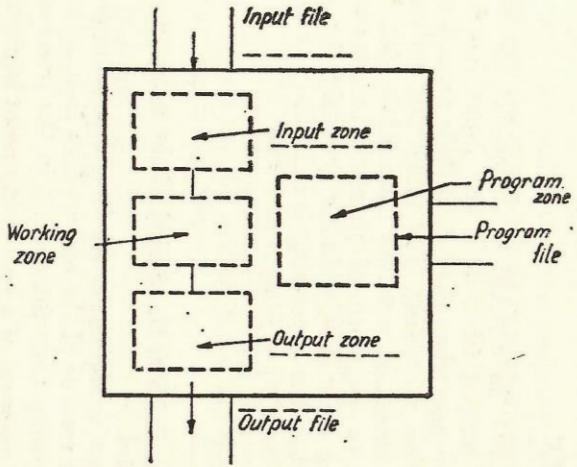


Fig. 3

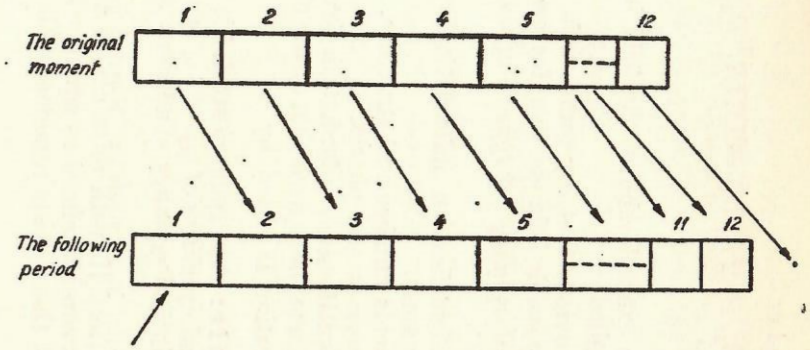


Fig. 4

There are two types of access to the data files:

a) sequential access;

b) random access.

The information processing is of two kinds:

a) — computation, when the current record in the output file is the result of some computations performed on the input records:

b) transfer when the current record in the output file is the result of certain record transfers in the input files.

2.5. Types of data file processing

When defining all the processing methods, the notation:

$$S_e := A(S_i) \tag{10}$$

is used, where

S_e = output file(s)

S_i = input file(s)

The following processing methods have been defined:

merging, conversion, adding, filing, break-down, updating, sorting, cumulation; a symbol has been attributed to each processing type. The conversion is described as the information transfer from a file of a certain type of medium to another file of a different type of medium, the files being therefore logically equal; only the medium codes are different. The operation is noted:

$$S_e := \underset{\text{BM}}{conv}\,(S_i)\ \underset{\text{CP}}{} \tag{11}$$

where

BM = magnetic tape code;

CP = punched card code.

Another example cited here is the break-down (marked as "def") performing a period conversion:

$$S_e\left(\frac{P}{n}\right) := def\,(S_i(P) \tag{12}$$

2.6. Types of memories

Permanent memory. According to the given definitions, a permanent store (memory) is characterized by:

$$S_e := S_i \tag{13}$$

Updating memory. According to definitions given above, the characteristic of an updating memory is:

$$S_e \simeq S_i \tag{14}$$

Dynamic (sliding) memory. The successive replacement of the stored information files, in a fixed number of periods.

Example: The working out of an account file for 12 months. When a month is over the former information is cancelled, the relative periods are shifted and the achievent on the current period is added (figure 4).

2.7. Types of data files

The data files are of three types:

a) *uniform files*, composed of records of identical length and structure;

b) *non-uniform files*, not satisfying the former conditions;

c) *partially uniform files*, consisting of sub-sets of uniform records.

All the uniform sections of a file can be minimized, as an information volume, as follows:

a) by computing the common factoring records;

b) by introducing only the data into the file, supposing that their significance is known owing to an invariable order of data files, to which implicit order numbers, instead of indexing files, may be attached.

2.8. Information tables

Table making represents the operation of working out an information table. Some additional definitions are required in order to make possible to define this operation:

A chain of identifiers is the specification of a file of identifiers, conventionally definig a break-down hierarchy. For example:

$$L(I1, I5, I0, I3)$$

where each identifier may have two values: *a* and *b*, representing the break-down hierarchy.

NOTES

1. The break-down is incomplete when certain value combinations are not allowed; their list is associated in the specification of the identifiers chain, e.g.:

$$L(I1, I5, I0, I3); N(ab\ xx, bb\ xx\ .....)$$

where x denotes the interdicted branches.

2. A chain defined as above is called of AND type, because every branch is specifiable by logical AND operations; sometimes it may be necessary to specify the AND — OR type of break-down hierarchy (with sub-totals), e.g.

$I1 = a \vee b$	$I5 = a \vee b$
$I1 = a \vee b$	$I5 = a$
$I1 = a \vee b$	$I5 = b$
$I1 = a$	$I5 = a \vee b$
$I1 = b$	$I5 = a \vee b$
$I1 = a$	$I5 = b$
$I1 = b$	$I5 = a$
$I1 = b$	$I5 = b$

It is obvious that, in this case, the break-down branches, additional to the chain, can be obtained from the chain branches by *OR* operations: that is why the following definitions are applicable only to *AND* chains.

Information table, defined as follows:

— each co-ordinate (x, y) is a final branch of a break-down hierarchy (L_x, L_y)

— a cumulating rule (Rx) is associated to every column (X). The vector of this rule is (R_x).

— the logical relationship Cxy denotes the belonging to a table.

The table making consists of the cumulating of a file on a statistical table, by:

a) identifying the case the current record belongs to, according to chains L_x, L_y:

b) cumulating according to (R_x) rules; particularly, the cumulating by means of counting, adds a given value $+ N$ (for example $+1$) in the (x, y) case, the table making operation is written:

$$S_T = \text{vent } (S_i) \qquad (15)$$

$$L_x,\ L_y,\ R_x,\ C_{xy}$$

Centralizing is a cumulation of the logically equal data in tables with the same structure, into a resulting table whose structure is identical to that of the two original tables. The operation is written:

$$S_T = \text{centr. } (S_{t_1} \ldots S_{t_n}) \qquad (16)$$

NOTES

1. The L_x, L_y chains from S_T and $S_{t_1} \ldots S_{t_n}$ are identical;

2. The (x_i, y_i) case represents the cumulating of

$$(x_i, y_i) \text{ cases of } S_{t_1} \ldots S_{t_n}$$

Types of tables

Fixed structure table, where the L_x and L_y represent functions. The fixed-structure table does not imply special requirements as regards the input data file.

Basic table, a fixed-structure table, without sub-total or sub-total files or lines.

Uniform table, a basic table, where the identifying indicators assume all the possible values and there are all the originated lines and files.

Linear table, a particular case of fixed-structure table, where the L_y chain includes only an identifying information, taking successive values.

Corollary. If the input data file is sorted according to the identifying information in L_y, the lines of the linear table can be successively obtained.

Uniform linear table, linear table, without sub-total, respectively total lines; the identifying information L_y takes all the possible values successively.

Combined tables, tables characterized by the fact they may be regarded as linear, in comparison with the basic tables.

The working-out of tables, the records of the output table file are obtained from the input file (files) records. For the fixed-structure tables, the values of the table boxes are obtained only after reading all the input file in the processing point; it follows that the fixed-structure table must be stored at the processing point. For the linear tables, supposing that the input file is sorted or a random access to this file exists, it suffices to store a small number of lines.

2.9. Types of information processing systems

There are two types of information processing systems:

a) integrated data processing systems, where the data file appears only once as an input at an information system processing point within a given period;

b) conventional information processing systems where the above conditions are not satisfied.

2.10. Types of problems for studying the information systems

It is estimated that three problems are associated to the study of the information system:

a) *The analysis of the information system* consists of the precise description of a given information system, requiring its defining at least at the level of information processing diagram.

b) *The rationalizing of the information system:* consists of the simplification of a given information system, without modifying the existing processing equipment or the algorithms used.

One of the following operations is at least supposed liable to perform the rationalization:

— elimination of files (records);
— the minimizing number and/or volume of memory files;
— reducing of the file identifying information;
— condensing of the processing sequences.

The main object of SIS-1 and SIS-2 programs is to rationalize the information systems.

c) *The synthesis of the information system* consists of the defining of the most economical input files and processing sequences, required for getting some defined output files, with certain optional limitations regarding the types and performances of the processing equipment and/or regarding the input data.

The SIS-3 programs are directed towards the synthesis of information systems.

3. THE PROCESSING STEPS, ACCORDING TO THE SIS METHODOLOGY

This section deals briefly with the working stages, showing some simplified examples. In practical work, standardized documents are used.

3.1. The process analysis (SIS-1)

The object of this step is to establish if all the dynamic conditions regarding the control of the analyzed process are satisfied (there is sufficient time for making decisions, in comparison with the data processing times, disturbances transmission laps and for process inertia, taking into account the stocks too), and to suggest possible required improvements.

In order to achieve this aim, the various possible control circuits for compensating the disturbances are analyzed, relationships of the following kind being verified for each circuit:

$$T_s = T_z - T_d - T_c - T_p \tag{17}$$

where

T_s = the possible equivalent stocking time
T_z = the disturbance mean period
T_d = the decision making time
T_c = the control inertia time
T_p = the process inertia time.

When no stock exists ($T = 0$), the relantionship is:

$$T_d \leqslant T_z - T_c - T_p \tag{18}$$

actually expressing the fact that the available time for making decisions is the difference between the mean duration of the disturbances that have to be compensated and the process inertia plus the control inertia.

The analysis of the economic process is performed as follows:

Step 1: the working out of the economic unit diagram;
Step 2: the identification of disturbances;
Step 3: the specifying of the various dead times;
Step 4: the identification of possible control circuits;
Step 5: analysis of the conditions (17, 18) for every control circuit;
Step 6: determination of the best control circuits.

3.2. The analysis of the information system (SIS-2)

The object of the analysis is *to ascertain* the existing information system in the unit considered, by means of identifying the data files and their processing, in order to establish the information flow chart, that will serve as a basis for system improvements, as well as to obtain the priorities for the operations liable to be passed on the electronic computer, from both *technical* (increasing the processing speed) and *economic* (a more reduced equivalent cost, compared with the conventional processing) points of view.

The information system is accurately described, so that the making up of every file is established per processing points and periodicity.

A graph is drawn for each period, based on the description data, in order to establish the processing flow chart; sequence numbers are then successively given, according to the order of obtaining the files one after another, according to the graph. All the files representing inputs or memory in the analyzed information system receive zero sequence number. After getting a primary representation of the present processing sequences in order to reduce their number, an analysis successively supposing the files as being memory is initiated. Several variants are obtained, out of which the one that decreases the processing time by decreasing the number of necessary file processing sequences is selected, even if some files become memories (possibly requiring organizational modifications). Of course it is possible to impose another criterion, for example to obtain a minimum number of memories.

The steps of the analysis are as follows:

Step 1: The unit and its activity are described by means of drawing a very simplified diagram of the information system and the processing points, as well as the operations inside the economic unit are defined.

Step 2: For each activity and each processing point, the origin or destination point of each file, and for every output file the corresponding input files are specified. At the same time, the time characteristic (period and phase) are specified for each file, so that necessary time conversion operations become apparent.

Step 3: For each period individually, the following sub-steps are performed to the file processing flow chart:

1. all the output files corresponding to a single period are separated, the corresponding input files being specified;
2. the corresponding file processing graph is established;
3. the graph is analyzed;

Step 4: The new performances obtained are analyzed (from the viewpoint of accelerating the work, costs, etc., using empirical formulae).

An example of standardized form, obtained in step 2, is given in Appendix 1.

3.3. **The information system synthesis (SIS-3)**

The problem consists in establishing the minimal information to be contained in the input files in order to get the output data files, and in drawing the file processing flow chart and the computing flow chart. The synthesis starts with the study of the final document.

The steps are as follows:

Step 1. The document that must result is studied, in order to ascertain the lines and columns; the user gives all the details and explanations regarding the logical checking conditions, computing rules, definitions of indicators or any other details the analyst may require (Appendix 2).

Step 2. The analysis of the lines and columns, in order to establish the data identifiers and their associated data, is performed (Appendices 3 and 4) and the list of the symbolized indicators and corresponding codes is drawn

up (Appendix 5). The affiliation criterion, as well as the affiliation rules to columns and lines are established (Appendix 6). Minimum input files result.

Step 3. Considering the explanations (step 1) and the minimal files (step 2), the file processing flow chart is established, showing the type and number of the required programs (Appendix 7).

Step 4. The following sub-steps are performed for each program established in the flow chart (step 3):

1.a — description of the way of obtaining the output records from the input records according to the restrictions imposed by the output document and to details in step 1;

b — a list of logical conditions is drawn up and a code is attributed to each logical condition;

c — a list of sub-routines and their description is drawn up;

d — a list of all the files used within a program and their codes is drawn up;

2.a — a graph, including all the input-output records, the logical conditions and the sub-routines used is drawn (Appendix 8);

b — sequences are associated to the graph;

c — sequence numbers are re-written on the record valuation list and on the logical condition list (Appendix 9).

3. a — The program is written in a conventional language, every instruction on an individual line satisfying the following priorities and rules.

A) Priorities

1. column text reading
2. column text printing
3.1. initial conditions
3.2. record reading
3.3. affiliation criteria checking
4. logical control
5. computations
6. final conditions
7. end of file printing

B) Rules

1. every input file must be read; the data access may be sequential or random. (The access in the considered examples was sequential);

2. the texts that are not associated to certain input records, sorted at least after an implied current number, may be originally printed;

3. until the reading of the input end of file, record processing at the record level is cycled;

b — the conventional program is minimized, by eliminating the repetitive logical conditions or certain redundant instructions.

4. Program flow chart drawing. The flow chart is subsequently coded.

On the basis of the example considered in Appendices 1—9, the program was written in COBOL; the resulting table is given in Appendix 10.

Item	OUTPUT							
	For output and final documents			Appointed date	Code	Period	Processing point	Destination
	Number of lines	Number of columns	Number of boxes files					
1	2	3	4	5	6	7	8	9
1					30	360	01	01
2	4	6	6		250	1	01	04
3	40	6	6		40	1	01	01 04
4	300	6	1800		50	90	01	03
5	30	9	270	1, 11, 21	80	10	01E	01
6	200	10	2000	3	130	30	01E	01
7	3	5	15		220	1		02
8	3	20	60		230	1	02	02
9	40	20	800		20	1	02	01
10					110	360	03	03
11	600	6	3600		70	30	03	01
12					210	360	04	04
13	4	6	6		200	1	04	01
14	1	8	8		60	1	04	01

INPUT							Processing type				
Code file	Version	Period	Phase	Source point	Prepa-ring	Number of records	Logical checking	Table making, fixed structure	Table making linear tables	Centralizing	
10	11	12	13	14	15	16	17	18	19	20	21
30		360		01	MEP	100					
20		1		02		80	x				1
30		360		01	CONS	100	x				1
200		1		04		4			x		3
250		1		01		40			x		3
10		90		I		50		x			2
40	2	90 $\overline{1}$	0...-89	01	ARH	44		x			2
60	2	10 $\overline{1}$	0...-9	04	ARH	40			x		3
70	2	10 $\overline{30}$	0, 1, 2	03	DEF	300			x		3
10	2	30 $\overline{90}$	0, 1, 2	I	DEF	50		x			2
80	2	30 $\overline{10}$	0...-2	01	ARH	60		x			2
130		30	−1	01	COMPI	400		x			2
130		30	−12	01	COMP2	400		x			2
90		1		I		40	x				1
100		1		I		20	x				1
220		1		02		3					4
90		1		I		40	x				1
100		1		I		20	x				1
230		1		02		9	x				1
110		360		03	MEP	500					
50	2	30 $\overline{90}$	0, 1, 2	01	DEF	600		x			2
120	2	30 $\overline{1}$	0...-29	I	ARH	100		x			2
110		360		03	CONS	500		x			2
210		360		04	MEP	100					
250		1		01		40					4
210		360		04	CONS	100					4
40		1		01		44			x		3
140		1		I		30			x		3

	[1]	[2]	[3]	[4]	[5]
FAMILY BUDGETS STATISTICS	Total number of complete salaried families	of which:			
		without children		with unsalaried and single children	
		current year	previous year	under school age	7 years old and above
[1] TOTAL EXPENDITURE					
of which:					
[2] for FOOD:					
[3] a) towards state and co-operative sector					
— vegetable products					
— animal products					
[4] b) towards private sector:					
— vegetal products					
— animal products					

Specifications:

1. the same number of records for the expenditure is filled up for each family (if the expenditure is null, 0 is entered);
2. primary data regard all the families;
3. the complete families imply at least a husband and a wife;
4. under school-age up to 7 years;
5. the results are printed on pre-printed form;
6. new studied families appear every year (for the new families, 0 is entered in column 3)
7. the primary information is of a very good quality, the errors being accidental and non-significant.

Column number	Repetition	Level	Contents	Chain end	Identifier		Date
					Symbol	Code	
1	1	1	Expenditure of complete salaried families	*	FAM SUREX	1 1	CHELT
2	1	2	Expenditure of complete, salaried families without children, current year	*	FAM SUREX	1 1	CHELT
3	1	2	Expenditure of complete, salaried families, previous year	*	NRFAM(−360)	1−2000	CHELT(-360)
4	1	2	Expenditure of complete, salaried families, with unsalaried, single children (of pre-school age)	*	FAM SUREX COP	1 1 1	CHELT
5	1	2	Expenditure of complete, salaried families, with children of school-age, attending a school	*	FAM SUREX COP GV	1 1 1 2	CHELT

The following necessary input files result:
IN1 (NRFAM, FAM, SUREX, COP, GV (CHELT)
IN2M (NRFAM (CHELT (—360)

Appendix 4

Line number	Repetition	Level	Contents	Conditions
1	1	1	Total expenditure	(CHELT)
2	1	2	For food	NATC = 1
3 and 6	2	3	State sector (co-operative), or private sector	SECT = 1,2
4 and 7; 5 and 8	2	4	Vegetal or animal products	PROD= 1,2

The following file results:
FEL (NATC, SECT, PROD (CHELT)
Because IN1 and FEL files refer to CHELT for the current year, they may be combined, forming the file:
IN1M (NRFAM, FAM, SUREX, COP, GV (NATC, SECT, PROD (CHELT)

Appendix 5

The indicator and code list

Indicator	Symbol	Code	Significance
Family	FAM	1	complete family
		2	incomplete family
Source of livelihood	SUREX	1	wage earners
		2	other sources
Number of children	COP	0	without children
		1	with children
Age group	GV	1	up to 7 years
		2	above 7 years attending a school
Nature of expenditure	NATC	1	for food
		2	other expenditure
Sector towards which the expenditure was made	SECT	1	state and co-operative
		2	private
Nature of food products	PROD	1	vegetal production
		2	animal production
Expenditure	CHELT	9(5)	
Family number	NRFAM	1...2 000	item of family under study

C x y: FAM = 1 SUREX = 1

LINE								COLUMN				
Type	Number of Line	cxy	NATC	SECT	PROD	Level	Notes	1	2	3	4	5
1	1	x				1	Yo = 2					
2	2	x	1			2			Coop=0	Fam. No.	Coop=1 Gv=1	Coop= 1 Gv=2
3	3	x	1	1		3		Chelt	Chelt	Chelt (−360)	Chelt	Chelt
4	4	x	1	1	1	4						
4	5	x	1	1	2	4						
3	6	x	1	2		3						
4	7	x	1	2	1	4						
4	8	x	1	2	2	4						

Line original basis = Yo + 1 + (File of value associated to basis −1)

Number of relative lines for modifying the associated file value

LINE (1) = 1
LINE (2) = 2
LINE (3) = LINE (2) − 1 + (SECT−1).3 = 3; 6
LINE (4) = LINE (3) + 1 + (PROD−1) 1 = 4,5; 7,8
in this case:
LINE (3) = 3. SECT
LINE (4) = LINE (3) + PROD

NOTE: The bordered section represents the repeating group of lines, differing from level 3 identifier value, that is SECT.

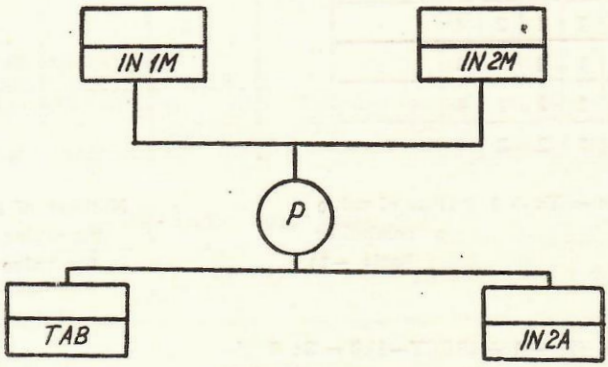
Input files

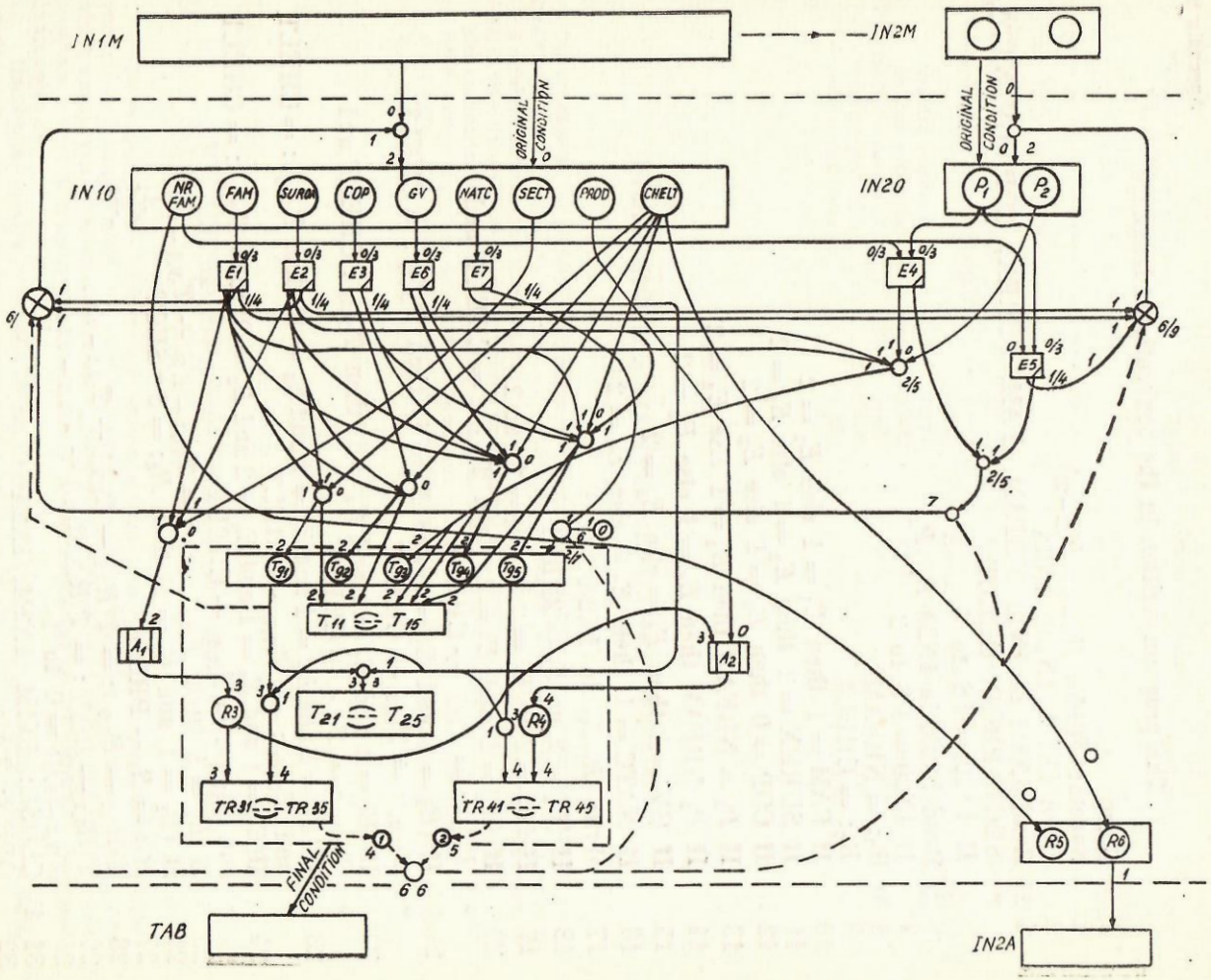
IN1M (NRFAM, FAM, SUREX, COP, GV, (NATC, SECT, PROD (CHELT)
IN2M (NRFAM, FAM (CHELT (—360)

The family number (NRFAM) was added in order to enable the record locating between IN1M and IN2M.

The files are supposed to be sorted, in order to avoid additional restrictions regarding the type of file access.

File processing flow chart




IN1M
IN2M
ORIGINAL CONDITION
IN10
NR FAM
FAM
SURD
COP
GV
NATC
SECT
PROD
CHELT
IN20
P1
P2
E1
E2
E3
E6
E7
E4
E5
T91
T92
T93
T94
T95
T11 T15
T21 T25
TR31 TR35
TR41 TR45
A1
A2
R3
R4
R5
R6
FINAL CONDITION
TAB
IN2A

Appendix 8.

The program written in the conventional language

```
1     Start
2     Spaces: T(9,5), R(5−6), P((1−2))
3     Open IN1M, IN2M, IN2A
4 1)  Read IN1M in IN10 NRFAM, FAM, SUREX, COP, GV, NATC,
  *   SECT, PROD, CHELT
      If IN10 = FS to 4
6 2)  Read IN2M in IN20 P1, P2
7     If IN20 = FS to 3
8 3)  R5: = NRFAM
9     R6: = CHELT
10    If FAM = 1 then E1 = 1 else E1 = 2
11    If SUREX = 1 then E2 = 1 else E2 = 2
12    If COP = 0 then E3 = 1 else E3 = 2
13    If P1 = NRFAM then E4 = 1 else E4 = 2
14    If P1 NRFAM then E5 = 1 else E5 = 2
15    If GV = 1 then E6 = else E6 = 2
16    If NATC = 1 then E7 = else E7 = 2
17    Write R(5−6) in IN2A
18    If (E1 = 2 or E2 = 2) in 1
19    If (E1 = 2 or E2 = 2 or E5 = 2) at 2
20    If (E1 = 1 and E2 = 1) then T(9,1) : = CHELT and
  *   T(1,1) : = T(1,1) + CHELT
21    If (E1 = 1 and E2 = 1 and E3 = 1) then T(9,2) : = CHELT
  *   T(1,2) : = T(1,2) + CHELT
22    If (E1 = 1 and E2 = 1 and E4 = 1) then T(9,3) : CHELT and
  *   T(1,3) : = T(1,3) + CHELT
23    If (E1 = 1 and E2 = 1 and E3 = 2 and E6 = 1) then T(9,4) : = CHELT
  *   and T(1,4) : = T(1,4) + CHELT
24    If(E1 = 1 and E2 = 1 and E3 = 2 and E6 = 2) then T(9,5) : = CHELT
  *   and T(1,5) : = T(1,5) + CHELT
25    If E7 = 1 then T(2,1 − 2,5) : = T(2,1 − 2, 5) + T(9,1 − 9,5)
26    If (E1 = 1 and E2 = 1) then R3: = 3 * SECT      [A1
27    R4: = R3 + PROD                                 [A2
28    T(R3,1 − R3,5 ): = T(R3,1 − R3,5 ) + T(9,1 − 9,5 )
29    T(R4,1 — R4,5 ) : = T(R4,1 R4,5 ) + T(9,1 − 9,5 )
30    T(9,1 − 9,5): = 0 to 1
31 4) CLOSE: IN1M, IN2M, IN2A
32    OPEN: TAB
33    TAB: = T8,5
34    STOP
```

FAMILY BUDGETS STATISTICS	Total number of complete salaried families	of which:			
		without children		with unsalaried and single children	
		current year	previous year	under school age	7 years old and above
TOTAL EXPENDITURE					
of which:	9,916	5,746	3,300	2,000	2,170
for FOOD:	8,309	4,733	2,470	1,486	2,090
a) towards state and co-operative sector	6,329	3,683	1,250	1,256	1,390
— vegetal products	3,327	1,987	690	20	1,320
— animal products	3,002	1,696	560	1,236	70
b) towards private sector:	1,980	1,050	1,220	230	700
— vegetal products	630	100	780	130	400
— animal products	1,350	950	440	100	300

ETUDE SUR LES SYSTEMES INFORMATIONNELS

Cet ouvrage se rapporte à un système de programmes en cours d'élaboration, destiné tout spécialement à l'automation des travaux statistiques courants. Vu que les éléments théoriques qu'il a fallu élaborer sont plus généraux, on a considéré aussi des applications se rapportant à divers types de systèmes informationnels, non seulement à ceux des systèmes statistiques.

Cet ouvrage comprend l'exposé général de la méthode, illustrée par des exemples, constituant en même temps une base de discussion pour un échange d'opinions entre spécialistes.

Il est prévu d'élaborer par la suite les spécifications des programmes ainsi que la manière de leur utilisation.

ИССЛЕДОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

Настоящая работа относится к системе программ в курсе разработки, которая имеет специальное направление по автоматизации обработки текущих статистических работ. Имея в виду, что теоретические элементы, подлежащие разработке, являются более общими, имелись в виду и применения к отдельным типам информационных систем, а не только статистических.

Настоящая работа содержит общее изложение метода и примеры, являясь и базой обсуждений в виду обмена мнениями между специалистами.

В продолжение этой работы предвидется разработать описания программ, а также и способ их эксплуатации.