A DATA BANK FOR ON-LINE PROCESS CONTROL

(The Synchrotron Injector)

A. Daneels

CONTENTS

	<u>Page</u>
1. INTRODUCTION	1
2. GENERAL	2
3. CHARACTERISTICS OF A CONTROL ELEMENT	3
4. CHARACTERISTICS OF CONTROL VECTORS	7
5. DATA BANK STRUCTURE ON DISK	7
5.1 Control elements	7
5.2 Record length for group characteristics	8
5.3 Record length for individual characteristics	8
5.4 Control vectors	8
5.5 Estimate of the number of sectors on disk	9
6. SOFT-CODE	9
7. EXAMPLE OF CODING INFORMATION IN THE DATA BANK	10
7.1 Group characteristics of control elements	10
7.2 Individual characteristics of control elements	13
8. DATA BANK HANDLING SOFTWARE SYSTEM	13
9. CONCLUSIONS	17
REFERENCES	18

1. INTRODUCTION

The need for a data bank results from the complete computerization of the Proton Synchrotron injector process. This process was designed so as to allow on-line control of most of its variables and a complex software system was developed for multiple and simultaneous control. Emphasis was put on "interactive control", i.e. several operators may control in parallel part of the synchrotron injector through computer driven consoles. Three consoles, called MIDI-CONSOLES¹⁾, with purely alpha-numerical displays are dedicated each to a well defined part of the process, whereas a fourth one, with alpha-numerical and graphical display, may control at random any single (or grouped) process variable(s). Conflicting control has to be avoided and therefore a centralized information base or data bank was developed.

Broadly speaking this data bank is subdivided into two complementary parts: a very small one, which is permanently in core, and a large one (at present around 10 K), which is stored on disk.

The small core resident section contains information which is likely to undergo frequent changes: a "flag" called "BUSY-flag" indicating the parts of the process currently under control, and the starting address of a binary matrix containing all "status" information, such as "ready" or "not ready", "on" or "off" and "polarity" status of the process variables. This matrix is built up of a number of 16 bit words (so far 25 words) in which every single bit represents a status (ready/not ready, or on/off, or polarity). Random arrangement of status bits within the matrix eases the hardware aspect of collecting this status information from the various variables. This matrix is permanently in core as part of a surveillance program, which is imbedded within the MIDI-CONSOLE¹⁾ program and which scans the various status bits at regular time intervals (at every booster cycle).

Nothing more needs to be said about this core resident section of the data bank. The following pages give a detailed discussion of the characteristics of the disk-based section of the data bank.

2. GENERAL

The control of a process is in general implemented by a great many control variables of different types.

To vary any control variable one needs to have some information at hand, as to its current control value, whether this variable is controlled via, for example, a stepping motor, what is its possible maximum value, whether it is switched on or off, etc. It will be shown here how, in general, one needs around twenty such pieces of information which are called the characteristics of every control variable.

"Control variable" has been mentioned so far without indicating whether they are single control elements, or several control elements combined into one control vector. Throughout this paper distinction will be made (where needed) between control elements and control vectors, where the latter are defined as a linear combination:

$$\tilde{X} = \sum_{i=1}^n \alpha_i x_i, \quad (1)$$

where

$\tilde{X}$ is the control value of a vector,
 x_i is the control value of the i^{th} element,
 α_i is a weighting factor,
 n is the number of elements.

Any statement about control variables is valid for either elements or vectors.

Any program controlling all or part of the process (e.g. on-line optimization) needs to know the characteristics of its control variables, which therefore should be stored, for example, on magnetic disk.

A process such as the synchrotron injector has around 500 variables each characterized by 20 pieces of information, hence one will need space for around 10,000 pieces of information. This plunges us into the problem of data banks, data structure, information retrieval and updating.

It will be shown how a data bank for control variables may be given a simple structure²⁾

It will also be shown how to define every control variable by a single code, SOFT-CODE, for retrieving its characteristics.

Eventually a software system will be described, which allows easy retrieving and updating some of the characteristics without the need for direct access to the data bank.

3. CHARACTERISTICS OF A CONTROL ELEMENT

A control variable has been defined (see the Introduction) as either a control element or a control vector. This section will describe the characteristics of every single control element. The control vectors will be dealt with in a subsequent paragraph.

i) Control element's name

This is for the purpose of displaying the name of the control element on alpha-numeric displays as, for example, on the MIDI-CONSOLE¹⁾. Such a name is in general a mnemonic indicating to which group the control element belongs, for example, a vertical steering magnet in the injection line of the synchrotron injector I-DV, followed by an item number, e.g. the 7th one. This full name is thus

I-DV07 .

Consequently a full name will be a mnemonic of four alpha-numeric characters indicating the group followed by a 2-digit item number. This arrangement covers a large number of possible "names".

ii) Physical units

Once again for displaying the measured value of a control variable one prefers this value to be indicated in physical units: for example, a current, even though measured as a voltage over a shunt, should be displayed in A or mA; an angular position, even though measured, for instance, as a voltage, should be indicated in mrad, etc. In general two characters will be sufficient for defining the physical units.

iii) A BUSY indicator

Any control variable may be controlled from various places. This is particularly true in a multiprogramming environment. To avoid conflicts one needs a centralized indication to tell whether a control variable is already under computer control or not. This indication is available in a "flag" as part of the core-resident section of the data bank (cf. Introduction).

iv) Value of the least significant bit (LSB) at acquisition

Suppose one measures a current. In general this is done through a shunt resistor with a fixed value so that one actually measures a voltage which depends "linearly" (assume the shunt is linear) on the value of the current. The ratio voltage/current depends on the value of the shunt resistor. The analogue value of the voltage is now converted into a digital value, so that going back through all the various conversions one finds a definition of the LSB (i.e. digital value = 1) as a function of the physical unit e.g.

$$\text{LSB} = 44.6 \text{ mA} .$$

v) Position of the decimal point

In previous examples, the smallest value one could measure corresponded to the value of the LSB (digital value = 1) at acquisition. Therefore, the least significant digit in physical units is 10 mA or 0.01 A, and consequently if the measured value were to be displayed as a 5-digit number in A one would have to insert a decimal point in front of the second digit right justified: e.g.

$$170.15 \text{ A} .$$

vi) Value of the LSB at control

Having observed the current value of the control variable, for example, on the display, one decides now to increase or to decrease this value. Therefore, one introduces a positive or negative increment defined in its physical units. This increment therefore has to be processed by the computer and converted into the appropriate digital value before sending it to the digital control register of the control variable: this implies that one should define the physical value of the LSB at control (digital control value = 1). If, for example, the maximum value of a power supply is 15 A and it is controlled by a 10-bit digital register then, of course, the LSB is 14.6 mA.

vii) Maximum and minimum value

This has been mentioned implicitly in the previous section. One needs to be safeguarded against exceeding limits which may not be symmetrical (e.g. unipolar supplies).

viii) Type of control element

One distinguishes, in general, between three types of control elements.

a) A-type

This is defined as the control element whose control value results from the algebraic sum of its previous control value and an increment, for example

$$x_t = x_{t-1} + \Delta_t, \quad (2)$$

where t is the time of control.

b) S-type

This is defined as the control element whose control value results from an increment only (e.g. Stepping motors)

$$x_t = \Delta_t \quad (x_{t-1} \text{ always} = 0). \quad (3)$$

c) C-type

This is in fact an A-type control variable for which there is no acquisition (measurement) foreseen. This applies more specifically to the preset counters of the synchrotron injector: in particular their control value includes the selection of a proper timing clock.

ix) Indication for READY-NOT READY

Some control variables may be READY or NOT READY depending, for example on whether there is a power supply connected to it or not. This indication is supplied to the computer for most elements of the synchrotron injector.

x) Indication ON-OFF

This indication allows one to find out whether an element has been switched on or off. If it is found to be switched off the computer should, of course, switch it on before control can start.

xi) Indication POLARITY

For some control elements the polarity of the measurement may be supplied in a different piece of information.

xii) RESET

When an element is found to be NOT READY the computer may attempt to RESET this element.

xiii) Acquisition and control addresses

The process computer is connected to the process by a transmission system*) through which measurements are acquired and control values sent out. In order to "multiplex" the transmission to or from the proper control elements, every element is given two addresses, one for acquisition, one for control.

xiv) Current control value

In order to avoid non-linearities between control value and measured value and also to avoid errors in the acquisition (e.g. missing bits) one may decide not to compute the new control value as the sum of the previous measured value and an increment but as the sum of the previous control value and an increment. This implies that the computer has memorized what value is actually on the digital control register of any control element.

xv) "Reference" value

This is a control value which was found to be satisfactory in the past as it ensured a good working condition of the process. If the current control value proves to harm the working conditions of the process one wants to return to the "reference" values.

xvi) "Buffer"

This allows one to compare the working conditions of the process resulting from the "current control value", with the conditions resulting from the "reference value", without destroying the "current control value" which is stored in the "buffer".

So far the characteristics of a control element have been enumerated. However, in general, some of these characteristics will be common to various control elements. As an example, all vertical steering magnets of the injection line of the synchrotron injector upstream to the distributor are called I-DV, their value is displayed in A, and they are of the A-type etc.

However, the 7th member of this group will be called I-DV07, and has an individual address for acquisition and control, etc.

So in general one can state that points (i) to (viii), inclusive, will be common to a group of control elements, whereas points (ix) to (xvi)

*) The synchrotron injector uses the digital STAR system (Système de Transmission Adressé Rapide) developed by the Control Group of the CERN Proton Synchrotron Division.

will be specific to each element of this group. Hence we come to a one-level data structure with group characteristics and individual characteristics.

4. CHARACTERISTICS OF CONTROL VECTORS

The control vectors have been defined previously (see the Introduction). A control vector is characterized by:

- i) the number of its control elements: e.g. N,
- ii) N codes: each defining a control element,
- iii) N coefficients: (which need not necessarily be integers) to weight the control value of every one of its control elements.

5. DATA BANK STRUCTURE ON DISK

5.1 Control elements

The simplest method is to store contiguously the record representing the characteristics of group J, then the characteristics of all its M individuals, then the specification of group J + 1, etc. However the number of individuals M may be different from one group to another and retrieving the characteristics of one single control variable would require, in general, knowledge of two pieces of information: the address (on disk) of its group characteristics and its item number within this group.

This arrangement is also rigid and would not allow, for example, giving different group characteristics to the same string of control variables (e.g. give it another name when various control elements of one group are combined into a control vector).

Consequently, one is led to a more flexible arrangement by storing into two different files the group characteristics and the characteristics of their individuals

... GROUP(J), GROUP(J+1), ...
INDIV(J,1), ... INDIV(J,M); INDIV(J+1,1), ...

Each record of group characteristics includes a pointer to the address on disk of the individual characteristics of the first element of this group.

By arranging the record of group characteristics to be of fixed length (10 words) one is able to code both the address on disk of the group record of a control element and its item number into one piece of information. An example of such a coding will be given in a subsequent section and will be called SOFT-CODE (Software Code).

5.2 Record length for group characteristics

Points (i) to (viii) inclusive, of Section 3 have been defined as group characteristics. The "name" has been defined as four alpha-numeric characters indicating the group followed by an item number within this group. This item number may be introduced in the code, called SOFT-CODE which will allow retrieving the characteristics of the control variable. Therefore, it need not be stored on disk. Consequently, one will need in general two words of storage for the name of the group (alpha-numeric characters are coded two characters per 16-bit word).

Furthermore, the record of group characteristics will include a pointer to the record of its first element (i.e. item 0).

By appropriate coding of other information such as decimal point position, busy indicator, and value of LSB, one can keep the length of the group record down to ten words (cf. Fig. 1).

5.3 Record length for individual characteristics

Points (ix) to (xv), inclusive, of Section 3 have been defined as individual specifications. The total length of this record is effectively of 10 words, as is shown in Fig. 1.

5.4 Control vectors

Control vector characteristics are imbedded within group characteristics of control elements. One distinguishes a vector from an element by its SOFT-CODE (see following paragraph).

To preserve the 10 word modular structure a vector of N elements is subdivided into subsets of four elements. The first word contains the number of elements; then four codes, SOFT-CODES, define the elements followed by their corresponding weighting factors α . The last word points to the next

following subset and is equal to -1 (hexadecimal/FFFF) in the last subset. However, so far there are no such examples among the synchrotron injector control vectors.

5.5 Estimate of the number of sectors on disk

In the particular example of the synchrotron injector, an IBM 1800 process controller is used, with moving search head magnetic disks for mass storage. Every disk has 512 K of 16-bit words storage capacity, subdivided into 200 cylinders of eight sectors each. Every sector has a capacity of 320 words. Hence 32 group characteristic records may be stored per sector. The number of groups of control variables for the synchrotron injector is estimated around 256, which corresponds to eight sectors. Another 24 sectors store the individual characteristics.

6. SOFT-CODE

Every control variable is defined by a SOFT-CODE retrieving the record of its characteristics in the data bank.

The IBM 1800 process controller which is used by the synchrotron injector is a 16-bit machine; consequently the SOFT-CODE is a 16-bit word which is organized as follows

i) Most significant bit, sign bit (MSB)

This bit is set to 1 for control vectors and is set to 0 for control elements.

ii) Group number (G)

In the case of the synchrotron injector 256 groups have been estimated; hence eight bits.

iii) Item number (I)

The remaining seven bits indicate the item number of a control element within its group.

This SOFT-CODE is effectively a pointer to the address on disk where the characteristics of its corresponding control variable are recorded. This address on disk is computed as follows:

- address of the record of group characteristics =

$$S_0 + Q \left\{ \frac{G \cdot 10}{N_S} \right\} + R \left\{ \frac{G \cdot 10}{N_S} \right\}, \quad (4)$$

where

S_0 is starting address on disk of the data bank,

G is the group number,

Q is the quotient of the division between braces,

R is the remainder,

N_S is the number of words per sector (= 320 in the case of the IBM 1800);

- address of the record of individual characteristics =

$$S_0 + P + (I \cdot 10), \quad (5)$$

where

P is the pointer to the first item of the group (first word in the group characteristics).

I is the item number.

7. EXAMPLE OF CODING INFORMATION IN THE DATA BANK

7.1 Group characteristics of control elements

i) Pointer "P" to the individual characteristics of the first item:

8 most significant bits = sector number Δ_S from S_0 on

8 least significant bits = number K of the 10 word record, within this sector, which contains the individual characteristics of the first element (item number 0) with $0 \leq K \leq (N_S/10)-1$.

Hence the address of the record of individual characteristics, which was defined in expression (5), may be rewritten as follows

$$S_0 + \Delta_S + (K + I) \cdot 10 . \quad (6)$$

ii) Control element's groups name (4 alpha-numeric characters):

is coded, for example, in ASC II, two alpha-numeric characters per 16-bit words (2 words).

iii) Physical units (2 alpha-numeric characters):

also coded in ASC II (1 word).

iv) Position of the decimal point and BUSY indicator:

8 most significant bits = BUSY which corresponds to 2 bits right justified per synchrotron injector zone. This coincides with the bit configuration of the "BUSY-flag" in the core-resident section of the data bank.

8 least significant bits = position of the decimal point, right justified in the displayed value.

A control variable is set to BUSY by setting the corresponding BUSY-bits, as they appear on the data bank, into a BUSY-flag located in the core-resident section of the data bank.

Example: the hexadecimal value /0302 means that the control variable belongs to the "injection line" zone and that a decimal point should be introduced in the displayed value (in physical units) two digits right justified.

- v) Coefficients: such as LSB at acquisition and control, and the coefficients which weight several control elements into a vector are not necessarily integral values and are coded as follows:

MSB: sign bit $\begin{cases} 1 = - \\ 0 = + \end{cases}$

2 bits divisor code $\begin{cases} 00 : \text{divide by } 1 \\ 01 \quad \quad \quad 10 \\ 10 \quad \quad \quad 100 \\ 11 \quad \quad \quad 1000 \end{cases}$

13 bits containing the modulus of the coefficient.

- vi) Maximum and minimum values: these are coded as normal binary values.

Example: In the case of a bipolar supply controlled by 10-bit digital registers, the maximum binary value will be 1024, and its minimum value -1024.

- vii) Type of control element: This could be coded in many ways. As an example we will show how this coding was performed for the synchrotron injector, so as to facilitate programming¹⁾.

Type	S	A	C		
			Timing clock		
			Magnet "B"	10 μsec	1 μsec
Indicator in data-bank	/0000 *)	/FFFF	/BFFF	/7FFF	/3FFF
After an exclusive-OR with /FFFF this indicator results in (not for S-type)					
		/0000	/4000	/8000	/C000

These are precisely the bits which have to be introduced in the control value of the preset counters to select the appropriate timing.

*) Hexadecimal values.

7.2 Individual characteristics of control elements

i) Indication for "Ready-Not Ready", "On-Off", and "Polarity"

A small program which is permanently in core reads continuously the "Ready-Not Ready", "On-Off" and "Polarity" status of all control elements (cf. Introduction). The result is stored into a matrix (also permanently in core) of let us say L binary words of each 16-bits (each bit representing a "Ready-Not Ready", "On-Off", or "Polarity" status).

Indications "Ready-Not Ready", "On-Off" and "Polarity" are coded as the coordinates of the corresponding status-bit within this matrix:

- 12 most significant bits indicating the row (binary word, 1 to 4095).
- 4 least significant bits indicating the column (position of the corresponding bit inside the word ranging from 0 to 15, right justified).

Control elements which do not provide this status are indicated by zero.

ii) Addresses for "Acquisition", "Control", and "Reset"

They depend on the transmission system between the computer and the process. In the particular case of the synchrotron injector, they are the STAR addresses.

iii) Reset control word

This is the "control value" (usually a binary word with only one bit on) which has to be sent out to the "Reset" address for resetting the control element which was found "Not Ready".

6.4 "Current Control Value", "Reference Value", "Buffered Value"

These are digital values as sent out by the computer.

8. DATA BANK HANDLING SOFTWARE SYSTEM

Every process control program has to refer to the data bank. However, one should prevent part or all of the data bank from being destroyed, for example by software bugs. In particular, the disks on which the synchrotron injector data bank is stored have a moving-search head and the

average access time is around 200 msec; moving the head can almost not be avoided because of the data bank structure (four cylinders or tracks; half of the first one contains group specifications, the remainder contains individual specifications). Therefore, some retrieving and updating operations need to be optimized to be short in time. As a general rule a programmer should not have direct access to the data bank, but only through an appropriate software system.

The various subroutines are written in machine language with FORTRAN compatible calling sequences. The error indicator is always zero after successful termination of the subroutine.

i) CALL SOFTC (NAME, ISOFT, IERR)

Given the NAME of a control element (six 16-bits EBCDIC coded alphanumeric characters) the subroutine SOFTC returns the corresponding SOFT-CODE into ISOFT. IERR is different from zero if no SOFT-CODE is found to match NAME (NAME not correct).

ii) CALL RBUSY (N, ISOFT, IERR)

Given the SOFT-CODE ISOFT of N control variables (elements or vectors), this routine checks their corresponding BUSY indicators in core. If this indicator is found to be already set to BUSY, the error indicator IERR is set to a negative value (e.g. -1), if not it sets the indicators to BUSY. IERR contains a positive value different from zero if any disk operation failed; it contains a negative value if a previous control program is controlling this variable. This subroutine should be called at the beginning of any on-line control program.

iii) CALL FRBUS (N, ISOFT, IERR)

Given the SOFT-CODE ISOFT of N control variables this routine resets the BUSY indicators. IERR is set to a value different from zero if any disk operation failed. This subroutine should be called at the very end of any on-line control program.

iv) CALL RDELM (N, ISOFT, ITAB, IERR)

Given the SOFT-CODE ISOFT of N control elements, all belonging to the same group, this program fills a table ITAB with the 10 group characteristics followed by $N \times 10$ individual characteristics (see Section 2). IERR is an error indicator for disk operation failure.

v) CALL WRELM (N, ISOFT, STAB, IERR)

This subroutine performs the inverse operation of RDELM, and updates the data bank with the contents of the table ITAB (10 group characteristics and $N \times 10$ individual characteristics).

vi) CALL RCOPL (N, ISOFT, ITAB, IERR)

This subroutine reads from the data bank, the vector characteristics (see Section 3) of N control vectors defined by their SOFT-CODE ISOFT and feeds them into a table ITAB of $N \times 10$ words in length. IERR has been defined previously.

vii) CALL WCOPL (N, ISOFT, ITAB, IERR)

This program performs the inverse operation of RCOPL and updates the data bank with the contents of the table ITAB ($N \times 10$ vector characteristics).

The subroutines RDELM, WRELM, RCOPL, WROPL read from or write into the data bank the characteristics of control variables all belonging to the same group. Retrieving and updating characteristics of control variables belonging to different groups is much faster if one proceeds in two steps: first retrieving and updating all group characteristics, next all individual characteristics.

viii) CALL RGRUP (N, ISOFT, ITAB, IERR)

This subroutine reads the group characteristics of N control variables defined by SOFT-CODE ISOFT and stores them in an $N \times 10$ word table ITAB (IERR, see above).

ix) CALL WGRUP (N, ISOFT, ITAB, IERR)

Performs the inverse operation and updates the data bank with the group characteristics of N control variables as provided in the table ITAB.

x) CALL RSGEL (N, ISOFT, IFRST, ITAB, IERR)

This subroutine is normally called after an RGRUP operation. It reads from the data bank the individual characteristics of N control elements defined by ISOFT and by the pointer to the first item within the group IFRST. This pointer is the first information contained in each set of group characteristics (see Section 5.3 and Fig. 1). The individual characteristics are stored in an $N \times 10$ word table ITAB).

xi) CALL WSGEL (N, ISOFT, IFRST, ITAB, IERR)

This subroutine performs the inverse operation.

xii) CALL READY (ICOOR, IERR)

This subroutine finds out the status of the "Ready-Not Ready" indication [see Section 3 (ix)] of a control element according to its corresponding coordinates (see Section 7.2). If this element is "not ready" the error indicator IERR is set to -1.

xiii) CALL ONOFF (I COOR, IERR)

This subroutine finds out the status of the "on-off" indication [see Section 3 (x)] of a control element, according to its corresponding coordinates (see Section 7.2). If this element is "off" the error indicator IERR is set to -1.

Both routines READY and ONOFF require a previous call to RDELM to read into core memory all control element characteristics, hence preventing redundant disk reading.

xiv) CALL CTVAL (ISOFT, IERR) (Control value)

This subroutine stores the current control value [see Section 3 (xiii)] of a control variable defined by ISOFT into the data bank. IERR is positive, different from zero in case of erroneous disk operation (see Fig. 2).

xv) CALL SRREF (ISOFT, IERR) (Store Reference Value)

This subroutine stores the current control value of a control variable defined by ISOFT into both the current control value location and the reference value location of the data bank (see Fig. 2).

xvi) CALL SNBAK (ISOFT, IERR) (Save Back)

The current control value is stored in the "Buffer" location, whereas the reference value is stored in the current control value location and also sent out to the control variable (defined by ISOFT) in the process (see Fig. 2).

xvii) CALL STBUF (ISOFT, IERR) (Set Buffer)

The value stored in the buffer location is introduced into the current control value location and sent out to the control variable (defined by ISOFT) in the process (see Fig. 2).

All those subroutines are modular for simple program structure as is shown in Fig. 3.

9. CONCLUSION

Implementing the data bank has proved very useful, especially during the commissioning of the various process components. For example, the calibration of the components required frequent modification of the definition of the least significant bit, both for control and for acquisition. This could be done in a flexible way through an interactive program displaying the current control variable specification on an alpha-numerical display and returning the modification to the data bank. All control programs have immediately the latest information in hand and need not to be reassembled and recompiled.

Furthermore only a rather small amount of information needs to be loaded in core for controlling part of the process.

However, the one-level structure is a limitation, particularly when vectors using similar elements are given different names for display purposes and also when various process variables only differ because of their name whereas all other group specifications are identical. In this case a two-level structure is more flexible but requires faster disk access.

Further improvement also should go towards full protection of the disk files containing the data bank.

Acknowledgements

The author gratefully acknowledges the discussions he has had with H. Van der Beken, L. Burnod, and W. Remmer, which have benefitted this work.

REFERENCES

- 1) E. Asseo, L. Burnod, A. Daneels and E. Sigaud, Système Midi-Console - Solution économique pour l'opération à accès rapide, via ordinateur, d'un processus à paramètres multiples, CERN/MPS-SI/CO 71-3. 27 Septembre 1971.
- 2) J.J. Duby, Data structures, Proc. 1970 CERN Computing and Data Processing School, Villa Monastero, Varenna, Italy, 30 August - 12 Septembre 1970, CERN 71-6 (1971), p. 335-404.

Figure captions

Fig. 1 : Data bank structure on disk.

Fig. 2 : Various controls provided by data bank software system.

Fig. 3 : Exemple of program using data bank system subroutines.

Distribution

List MPS-SI/1

and:

- R. Rausch	Lab. II
- J.T. Hyman	"
- M. Crowley-Milling	"
- F. Beck	"
- B. Sagnell	"
- J. Altaber	"
- G. Shering	"
- J. Bosser	"
- L. Burnod	"
- R. Bossart	"

Fig. 1

- READ PART OF DATA BANK FROM DISK INTO CORE
- INFORMATION FLOW BY "CALL CTVAL" IN SEQUENCE 1,2
- INFORMATION FLOW BY "CALL SRREF" IN SEQUENCE 1,2,3
- - - - - INFORMATION FLOW BY "CALL SFBK" IN SEQUENCE 1,2,3,4
- INFORMATION FLOW BY "CALL STBUF" IN SEQUENCE 1,2,3
- STORE UP DATED INFORMATION INTO DATA BANK ON DISK

Fig. 2

Fig. 3