TECHNICAL MEMORANDUM

134

Decoding data represented in FM94 BUFR

J.K. Gibson and M. Dragosavac

Operations Department

June 1987

European Centre for Medium-Range Weather Forecasts
Europäisches Zentrum für mittelfristige Wettervorhersage
Centre européen pour les prévisions météorologiques à moyen

1. INTRODUCTION

FM 94 BUFR is a computer independent binary form for the representation of meteorological data. Data are represented and defined using binary values which are not aligned on computer words, and which require some transformation before used by computer applications, including those applications designed to display the contents of the data. The term "decoding" is used to describe this transformation; the process could also be described as an expansion process, as it expands binary values into computer words, and expands the data description into a set of element descriptors.

2. OBJECTIVES

The decoding or expansion process defined below is intended to be general. Applicational implementation will vary from machine to machine with respect to:

- the storage and handling of tables
- The number of binary bits per computer word for internal representation of integer, real, character, logical and Boolean data
- the facilities available to read and write data
- the programming languages suitable for coding and executing the expansion processes
- the means by which a binary value is addressed and converted to a computer word or entity of type integer, real, character, logical or Boolean.

Nevertheless, a "generalised" but machine specific expanded form can be defined in terms of normal computer addressable entities such as typed variables and typed arrays. Thus the objectives of the specification for the decoding process become:

- the definition of the expanded form
- the specification of the expansion algorithms
- the implementation methods for applying the specified algorithms.

3. THE EXPANDED FORM

3.1 A generalised expanded form

If a standard expanded form of a sufficiently general nature is adopted then computer applications may interface to BUFR through this standard form. Since

the expanded form is expressed in terms of computer addressable entities, only the expansion process is machine specific; the expanded form is machine independent from the view point of its use by dependent applications. Thus, the expansion process produces data which is machine specific (with respect to representation), but which may be addressed by machine independent applications (with respect to variables, arrays, etc., at the programming language level).

To retain generality, the expanded form must:

- include details from each section of the BUFR message
- retain tabular information concerning the data description
- present data values in a suitably addressable form.

To achieve this, data should be expanded into a set of arrays.

3.2 The identification section

All data from this section (section 1) may be represented using type INTEGER, provided the flag information from octet 8 is transformed to an equivalent numeric value. The resulting integer array, ISEC1, will have magnitude IDIM1, and will contain:

SUBSCRIPT	OCTET	C O N T E N T S
1	1 - 3	Length of section (octets)
2	4	BUFR edition number
3	5 - 6	Originating centre
4	7	Update sequence number
5	8	Integer value of the octet containing flag bits (zero if section 2 is not present)
6	9	BUFR message type
7	10	BUFR message sub-type
8	11 - 12	Local table version number
9	13	Year
10	14	Month
11	15	Day
12	16	Hour
13	17	Minute
14-IDIM1	18 -	Reserved for local use.

3.3 The optional section

All data from this section (section 2) is assumed to be capable of being represented using type INTEGER. The resulting array, ISEC2, will have magnitude IDIM2, and will contain:

SUBSCRIPT	OCTET	C O N T E N T S
1	1 - 3	Length of section (octets)
2	4	Reserved (set to zero)
3-IDIM2	5 -	Reserved for local use

Note that where section 2 is not given, IDIM2 will be 1, and ISEC2 (1) will be zero.

3.4 The data description section

The data from the first 7 octets of this section (section 3) may be represented using type INTEGER, provided the flag information from octet 7 is transformed to an equivalent numeric value. The resulting integer array, ISEC3, will have magnitude IDIM3, and will contain:

SUBSCRIPT	OCTET	C O N T E N T S
1	1 - 3	Length of section (octets)
2	4	Reserved (set to zero)
3	5 - 6	Number of data sub-sets
4	7	Integer value of the octet containing flag bits

Note that IDIM3 will normally be 4.

The data from octet 8 onwards contain the data description. For decoded or expanded data, the data values may be defined by presenting:

- their names
- the units in which the values are returned.

Usually, this information will be extracted from BUFR table B. Where data are decoded from additional fields associated with other data as a result of the application of the "add associated field" operator, a name of "ASSOCIATED FIELD"

will be returned. The names and units may be represented by arrays of 40 character and 24 character entities respectively. Where values are returned as code figures or flag bits the corresponding units entity will contain "CODE TABLE NNNNNN" or "FLAG TABLE NNNNNN", with the correct table number substituted for NNNNNN. Where values are simply numeric or dimensionless, and in the case of associated fields, the corresponding units will be returned containing blank characters. The resulting character arrays, CNAMES and CUNITS, will have magnitude M, where M is the number of data values per sub-set.

3.5 The data section

The data from the first 4 octets of this section (section 4) may be represented using type INTEGER. The resulting integer array, ISEC 4, will have magnitude IDIM4, and will contain:

SUBSCRIPT	OCTET	C O N T E N T S
1	1 - 3	Length of section (octets)
2	4	Reserved (set to zero)

The data from octet 5 onwards contain the values defined by the data description. These values are converted to the units as specified in BUFR table B during the decoding. Values are by nature:

- real, if dimensioned values are represented
- integer, if code figures or dimensionless values are represented
- character, if specific identificational data or plain language data are represented
- bit patterns, if values refer to flag tables.

Since the bit patterns may be represented as type INTEGER by using an equivalent numeric value, and since INTEGER values may be preserved when represented as type REAL, all values except character values may be represented as type REAL. The resulting real array, VALUES, will have magnitude M by N, where

- M is the number of data values per sub-set
- N is the number of data sub-sets.

To represent data of type CHARACTER, a supplementary array of 80 character entities, CVALS, will have magnitude IDIMC. Where an item of character data is to be returned, the corresponding element of VALUES will contain a real number which, when truncated to an integer, will represent

ISUB * 1000 + ILEN

where:

- ISUB is the subscript of the element of CVALS at which the character value may be located
- ILEN is the number of characters represented.

Each character value will begin left justified at the start of an element of CVALS, and continue through as many elements as necessary, the final element being blank filled.

3.6 The data section expanded for display purposes

If the purpose of the BUFR decoding is to display the data (rather than to interface to a computer application), further expansion into a display form may be desirable. This involves resolving the code figures and flag values in terms of the appropriate character descriptions from the tables. To achieve this, code table or flag table entries are placed in CVALS, and the corresponding element of VALUES is assigned a pointer to the location and length of the character information, as defined in 3.5 above. Where this further expansion has been completed, the resulting form shall be referred to as the "display form".

3.7 Supplementary information

During the decoding process a list of supplementary information will be compiled for return to the user. Such information may be represented using type INTEGER. The resulting array, ISUP, will have dimension 16, and will contain:

SUBSCRIPT	CONTENTS
1	IDIM1
2	IDIM2
3	IDIM3
4	IDIM4
5	M
6	N
7	IDIMC
8	Total length (octets) of the BUFR message (including section 0 and section 5)
9 - 16	Reserved for future use.

3.8 Summary of the expanded form

The expanded form, as defined above, is comprised of a set of arrays as follows:

ARRAY	MAGNITUDE	C O N T E N T S
ISEC1	IDIM1	Section 1 information
ISEC2	IDIM2	Section 2 information
ISEC3	IDIM3	Section 3 preliminary information
CNAMES CUNITS	M) M)	Section 3 data definition
ISEC4	IDIM4	Section 4 preliminary information
VALUES	N by M	Section 4 data values
CVALS	IDIMC	Section 4 character data values
ISUP	16	Supplementary information (IDIM1, IDIM2, IDIM3, IDIM4, N, M, IDIMC, etc.)

4. THE EXPANSION ALGORITHMS

4.1 Overview

To achieve the expanded form as defined in 3 above, it is necessary to:

- expand specific octets into values of type INTEGER (ISEC1, ISEC2, ISEC3, etc.)

- obtain a decoded data description
- decode the data
- complete the list of supplementary information (ISUP).

These processes are illustrated in Fig. 4.1.

The expansion of octets contents is trivial, is likely to be influenced by machine related facilities, and is thus not relevant to this discussion. It should be noted, nevertheless, that a set of bit manipulating routines called GBYTE, GBYTES, SBYTE and SBYTES are frequently used by many centres, and are available for many machine implementations from the National Center for Atmospheric Research, Boulder, Colorado.

4.2 Decoding the data description

The decoding of the data description is intended to derive two sets of information:

- a working table, corresponding to the coded data
- lists of the names and units of the decoded values.

The working table is identical in format to BUFR table B. When complete, the working table contains one entry for each data field corresponding to one data sub-set. Thus for data represented without data compression, the first data sub-set may be identified by a one to one correspondence between each entry in the working table and each data field in the data. For data represented with data compression, each set of elements may be identified by a one to one correspondence with the entries in the working table (see Fig. 4.2).

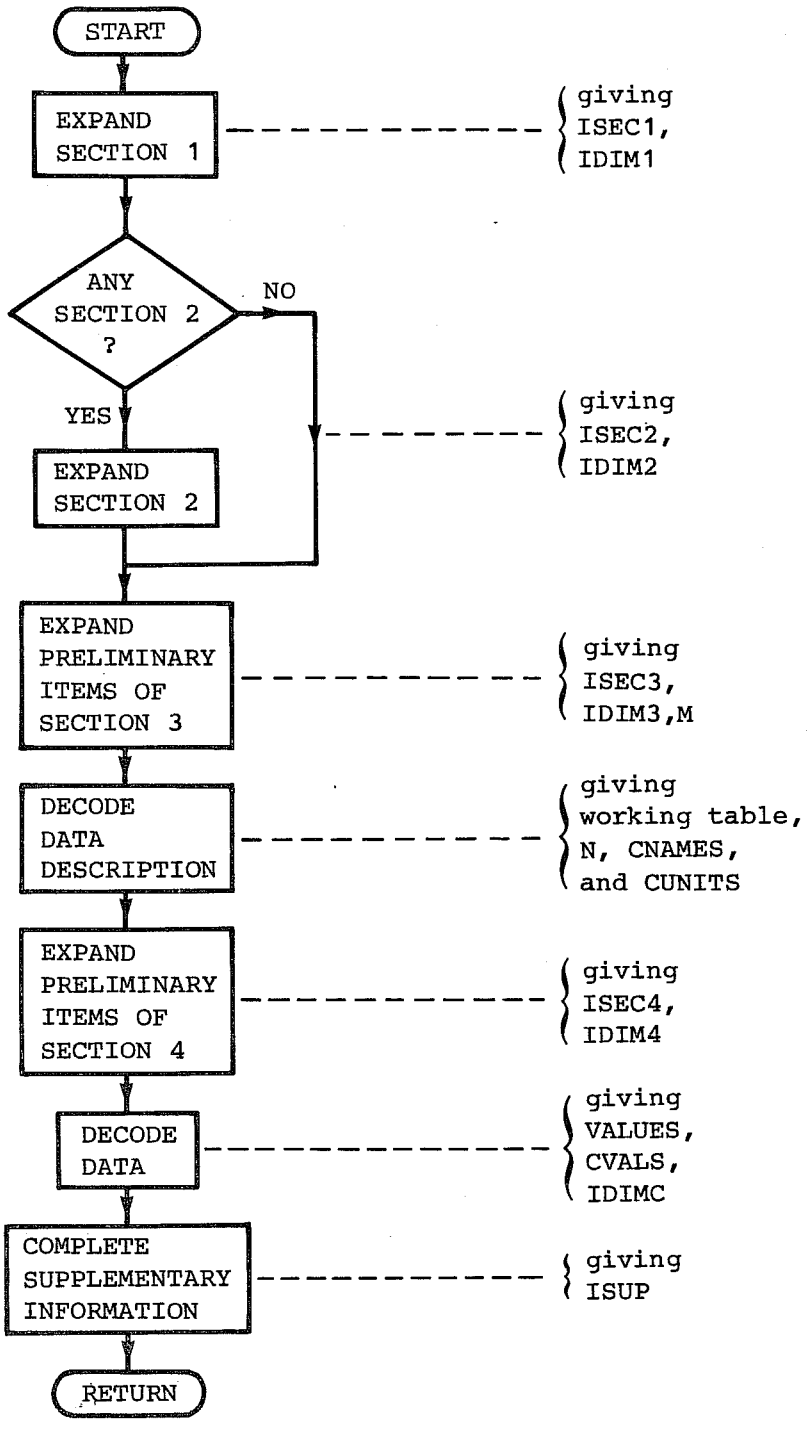

Fig. 4.1: BUFR Decoding

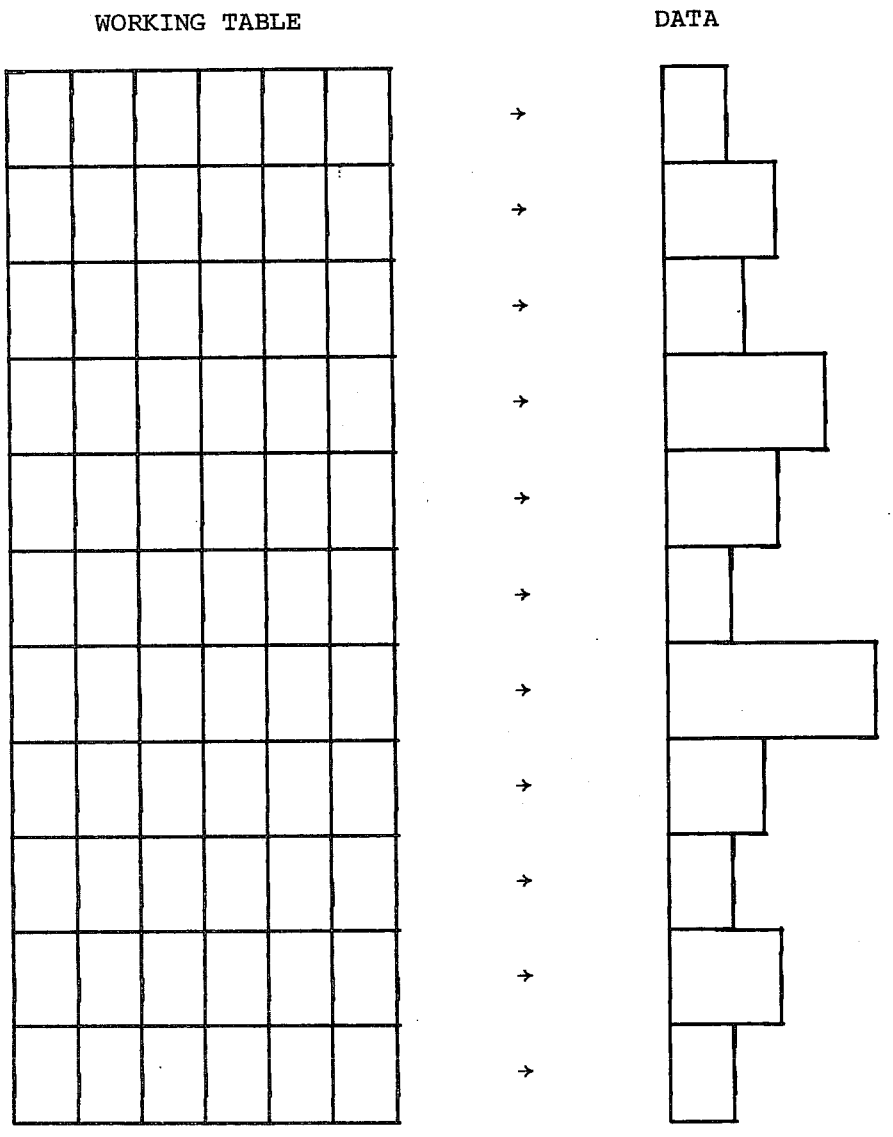

Fig. 4.2: Correspondence between Working Table entries and Coded Data

Where a table B element descriptor exactly describes a data field, the table B entry is copied to the working table unchanged. Where the table B entry is modified by the application of an operator descriptor, the appropriately modified entry is copied to the working table. Where the corresponding data item is an associated field, a special entry is added to the working table; this is comprised of an element name of "ASSOCIATED FIELD", a blank filled units field, and a data width corresponding to the data width of the associated field. The table reference and scale entries are ignored.

An additional special entry is added where element descriptors are used following a "change reference value" operator descriptor, to define changed reference values. In this case, an element name of "REFERENCE VALUE", a blank filled units field, and a data width equal to the Y value of the operator descriptor is used. The table reference and scale entries are ignored.

Where data compression is used, the working table entries each need to contain two additional items:

- data width of increments (IW)
- total data width of element set (IWS).

If IDW = data width of element as obtained from table B, and modified by any operator in force, and N = number of data sub-sets, then:

- the width of IW = 6 bits
- IWS = IDW + 6 + N*IW

Adding IW and IWS to each working table entry enables any part of the data to be located, accessed and decoded without the need to decode all of the data. This is particularly important when locating delayed replication factors and when detecting element sets of equal values (IW = 0).

The working table is not part of the expanded form - it is required to decode the data. The requirements of the expanded form are fulfilled by extracting matching entries for CNAMES and CUNITS to the decoded data values. The working table also enables selected elements to be decoded without the need to decode the complete set of data.

Fig. 4.3 outlines the processes involved in decoding the data description. They may be stated as follow:

a) get next descriptor; if none, go to z)
b) replication descriptor? If not, go to c)
c) perform "solve replication problem"
d) go to a)

e) element descriptor? If not, go to j)
f) perform "update working table"
g) does working table entry added correspond to data to be returned? If not, go to a)

h) update CNAMES, CUNITS,M
i) go to a)

j) sequence descriptor? If not, go to m)
k) perform "resolve table D reference"
l) go to a)

m) perform "process operator"
n) go to a)

z) return, giving working table, M, CNAMES and CUNITS.

Fig. 4.4 outlines the process "update working table". The steps involved are:

a) associated field present? If not, go to d)
b) non-units element descriptor? If not, go to d)
c) add special entry for associated field

d) search table of augmented table B entries for this element
e) augmented entry exists? If not, go to g)

f) get augmented entry
g) go to i)

h) get standard table B entry

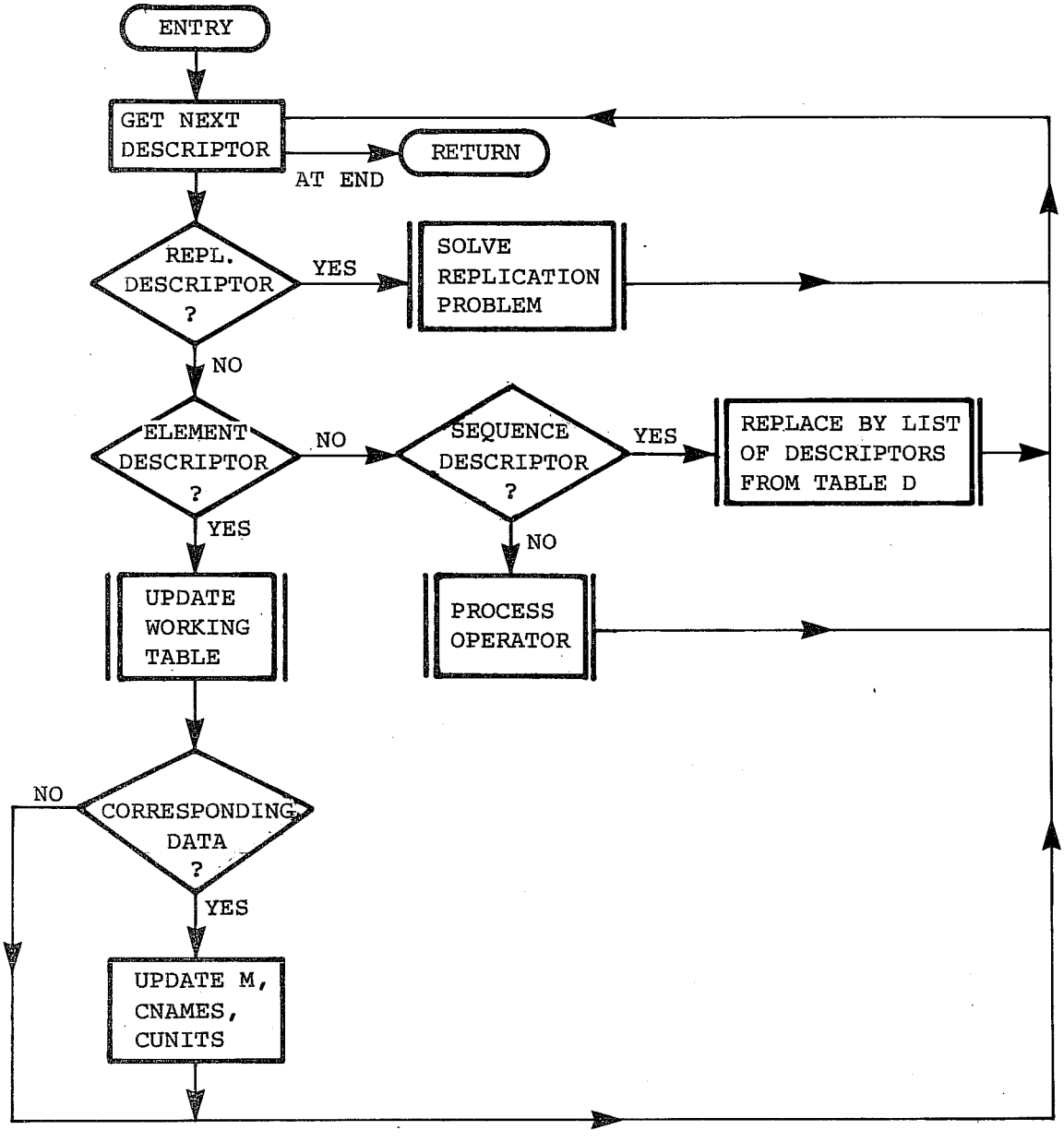

Fig. 4.3: Decode Data Description

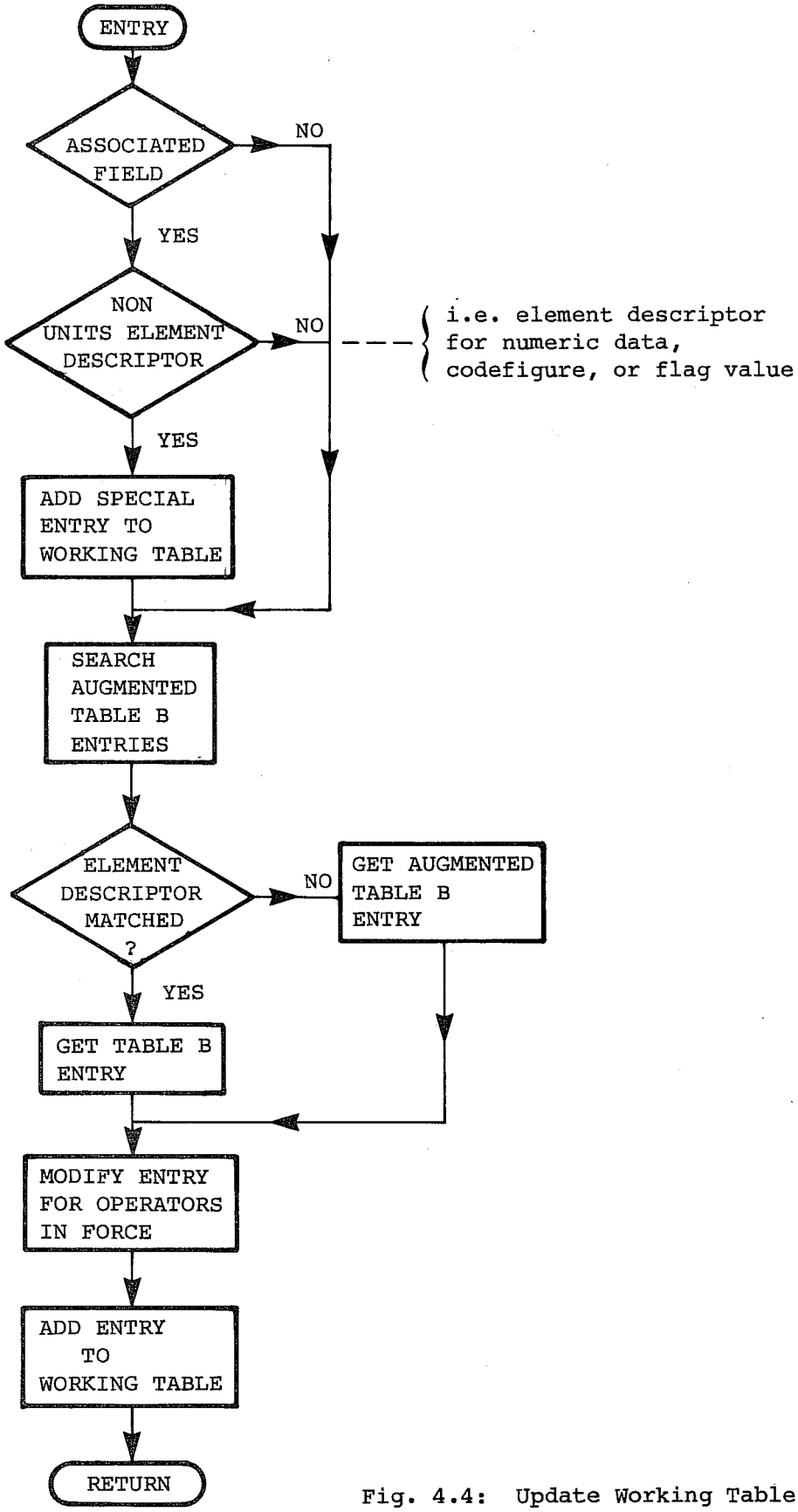

Fig. 4.4: Update Working Table

i) are further modifications required? If not, go to j)

j) modify entry for operations in effect

k) add entry to working table

z) return, with working table updated.

Some operator modifications will be such that an entry in an augmented table B will be required (e.g. where the reference value is changed). Other operator modifications can be applied to a standard table B entry (e.g. change data width, change scale) immediately before addition to the working table.

Fig. 4.5 outlines the process "resolve table D reference". The steps are:

a) obtain list of descriptors from table D

b) push down stack of descriptors forming data description, to make room for list

c) place list on stack

d) adjust stack pointer to get first descriptor from list as next descriptor

e) is replication in force? If not, go to z)

f) adjust counts of descriptors to compensate for list length

z) return, with list on stack and pointer set.

Fig. 4.6 indicates the processes required to set up replication control. They include:

a) store number of descriptors to be replicated, K

b) delayed replication? If not, go to g)

c) get next descriptor

d) replication factor? If not, go to K)

e) update working table

f) locate replication factor within the data

g) store the replication factor, J

h) get next K descriptors

i) replace by J copies of the K descriptors

j) go to z)

k) sequence descriptor? If not, go to n)

l) perform "resolve table D reference"

m) go to e)

n) return with error condition

z) return with replication completed

This approach resolves replication at the descriptor level. Each set of descriptors is replicated before resolving the table B and table D references. although this requires more table references in subsequent processing, it is logical and safe. It also provides a simple means for dealing with nested replication, as the outer nests are resolved completely before the inner nests are reached.

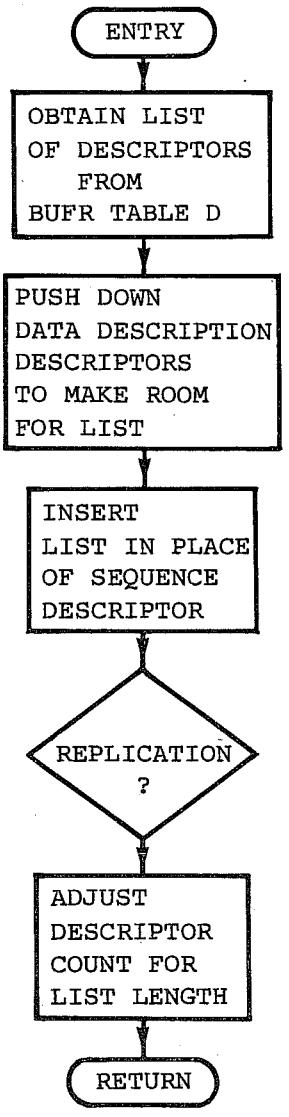

Fig. 4.5: Resolve Table D Reference

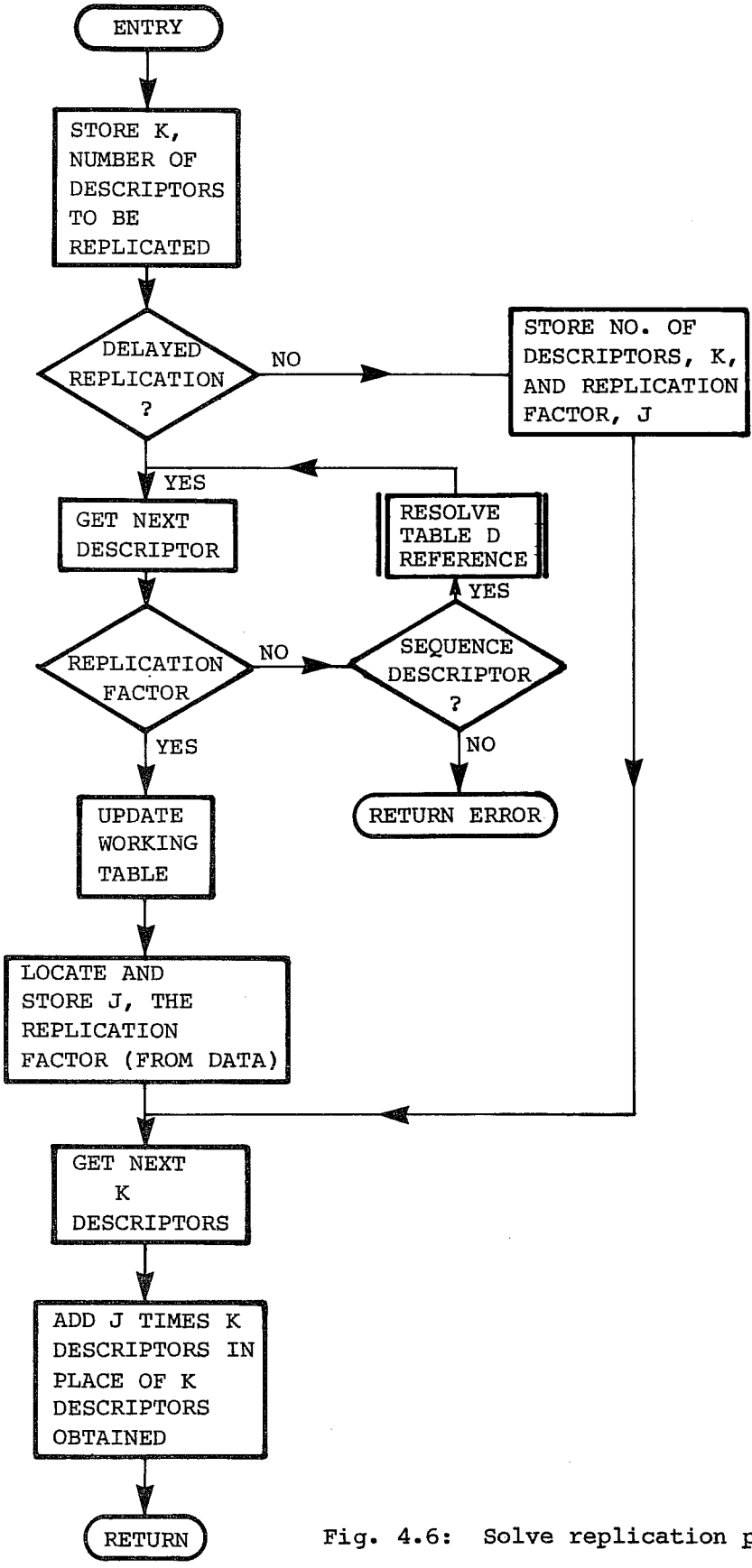

Fig. 4.6: Solve replication problem

Fig. 4.7 illustrates the actions required to process an operator. They include:

a) change data width) If not, go to d)
b) update constant containing data width increment
c) go to z)

d) change scale? If not, go to g)
e) update constant containing scale multiplier
f) go to z)

g) change reference value? If not, go to j)
h) process "update augmented table B"
i) go to z)

j) add associated field? If not, go to m)
k) update constant containing width of associated field
l) go to z)

m) signify character? If not, go to p)
n) add special entry to working table
o) go to z)

p) return with error condition
z) return with operator processed.

Currently, only one operator, change reference value, requires entries to be added or removed from an augmented version of table B. Fig. 4.8 indicates the processes involved:

a) is Y = 0? If not, go to e)
b) clear augmented table B
c) go to z)

e) get next descriptor
f) element descriptor? If not, go to k)
g) add special entry to working table

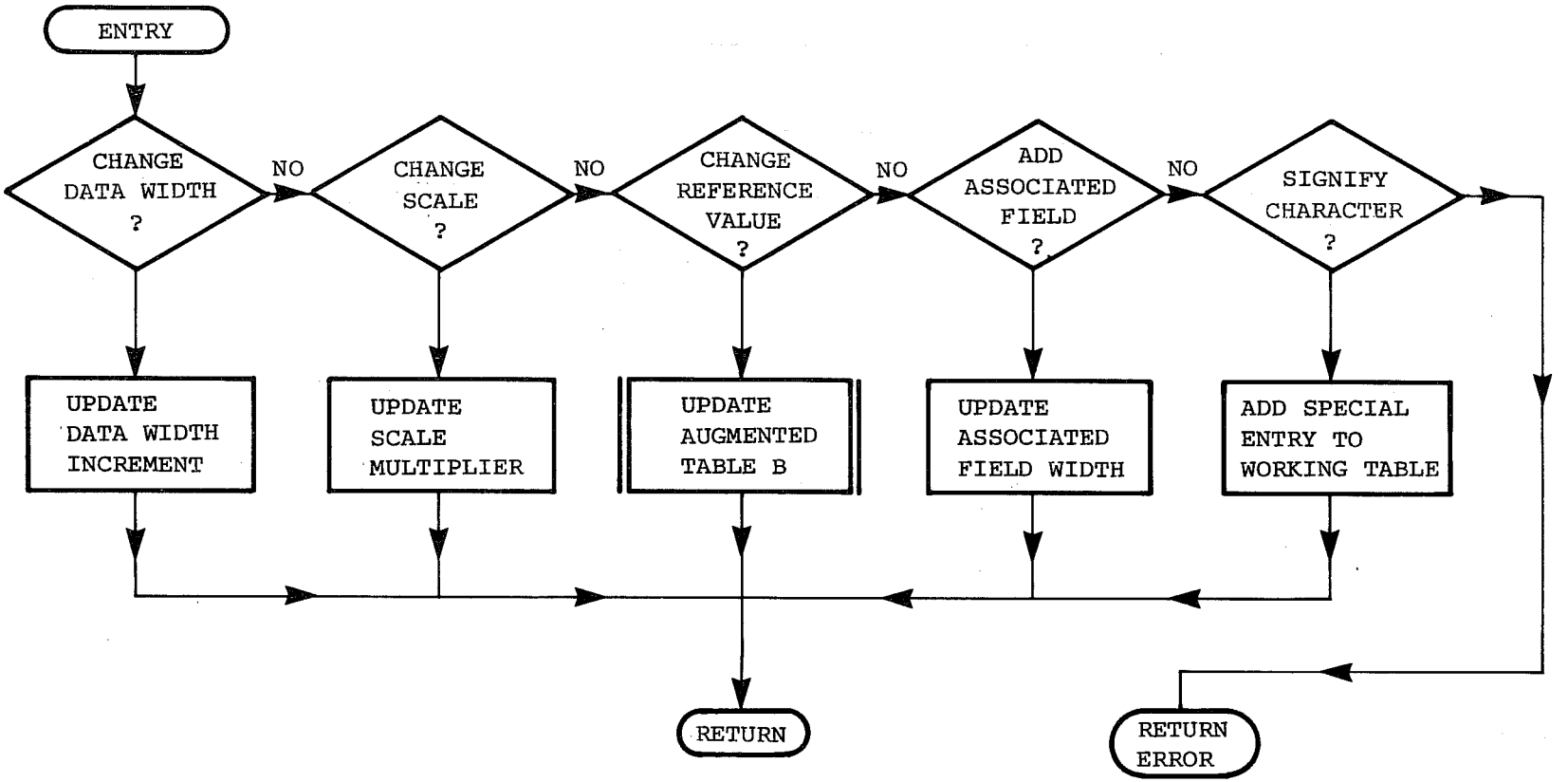

Fig. 4.7: Process Operator

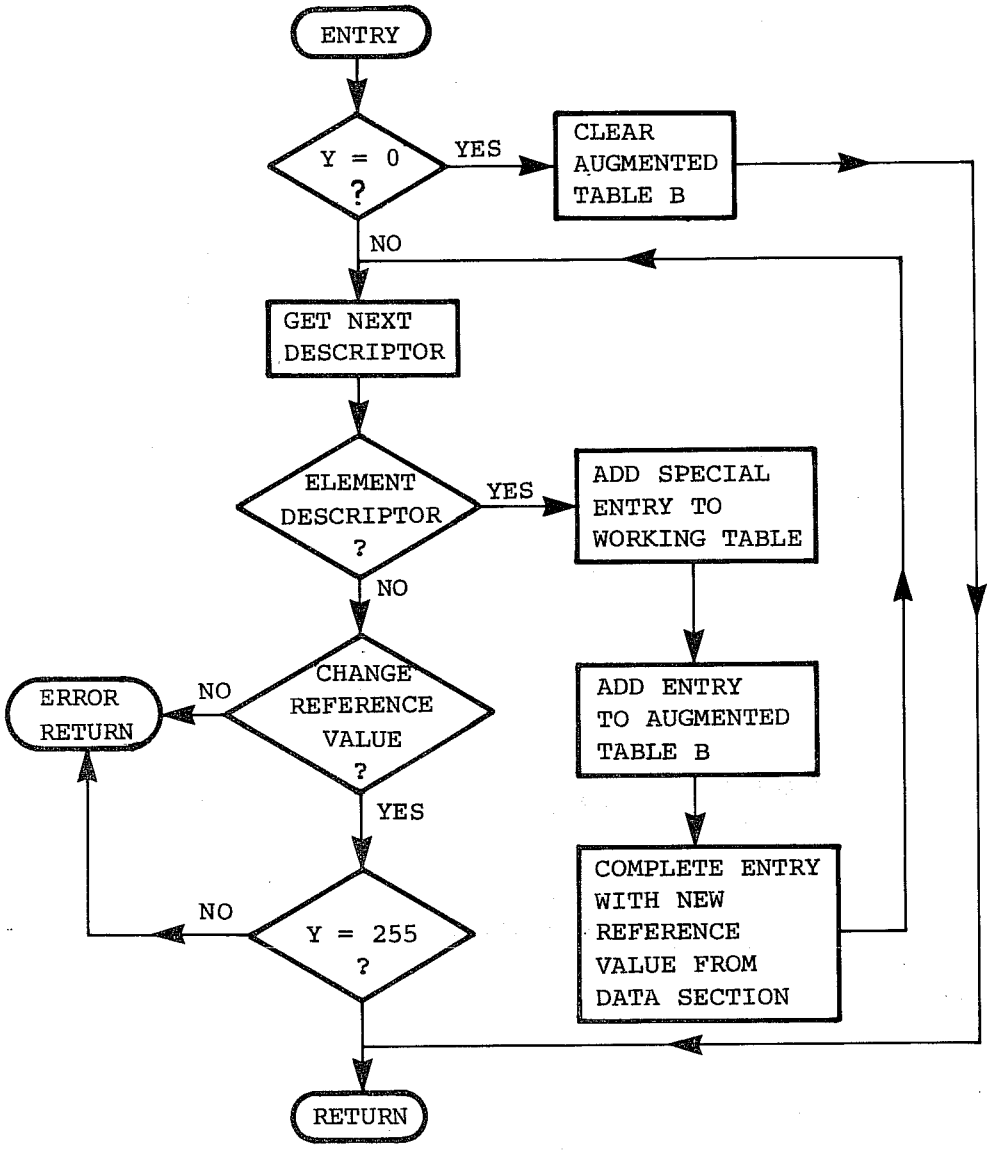

Fig. 4.8: Update Augmented Table B

h) obtain table B entry

i) obtain reference value from data

j) replace reference value

k) add modified entry to augmented table B

j) go to e)

k) change reference value? If not, go to n)

l) is Y = 255? If not, go to n)

m) go to z)

n) return with error condition

z) return with augmented table B updated.

4.3 Decoding the data

Having decoded the data description, the working table contains full details of the data. It would be possible from these details to locate data of a given name, or to decode the data completely. The description that follows relates to a complete decoding. A partial decoding would follow the same scheme, but would require the required elements to be matched by name.

The means of representing data values within the real array VALUES and within the character array CVALS is fully described in 3.5 above.

The processes involved in decoding the data are illustrated in Fig. 4.9, and include:

a) set pointers and counters to initial values

b) get next working table entry

c) special entry for new reference value? If not, go to e)

d) go to b)

e) use working table to locate data

f) data compression? If not, go to h)

g) expand to give N values

h) character data? If not, go to k)

i) move character data to CVALS

j) compute pointers to CVALS, to be stored in VALUES

k) go to p)

l) display form required? If not, go to p)

m) is data code figure or flag? If neither, go to p)

n) obtain character data from entries in code or flag tables

o) go to i)

p) update the array VALUES with value(s) or pointer(s)

q) update pointers, re-setting to re-use working table if required

r) more data? If not, go to z)

s) go to b)

z) return, with decoded data in CVALS and VALUES

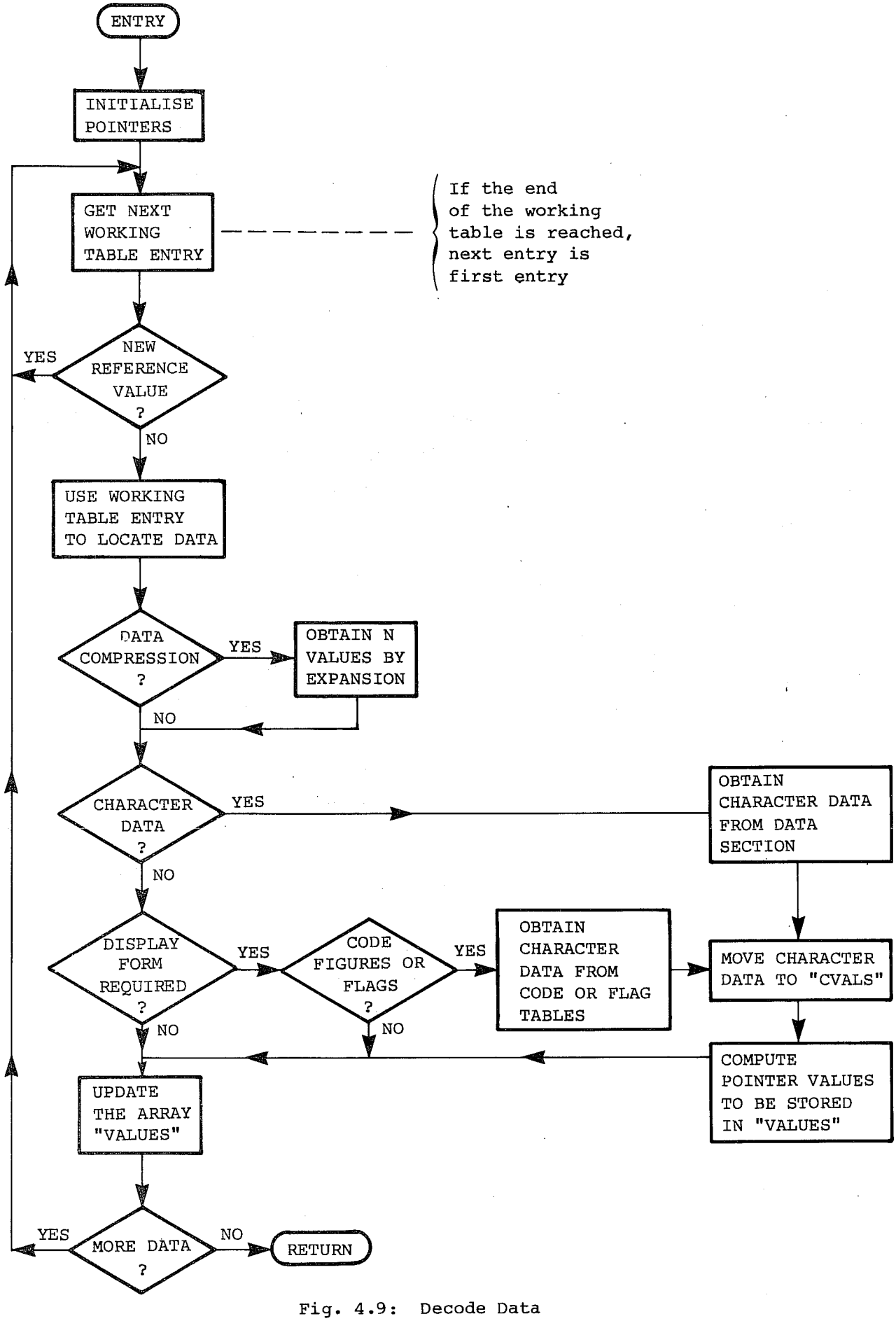

Fig. 4.9: Decode Data

Note that CNAMES and CUNITS could be extracted from the working table at step p) above. Note also that selective decoding requires:

- a list of required names as input
- a check at step e) above to determine whether this working table entry conforms with required data
- a branch to step q) if the data item is not required
- some care to retain associated fields with required items, by reference to the next entry if an associated field is encountered.

5. IMPLEMENTATION METHODS

The implementation methods adopted will differ from centre to centre. Due consideration will need to be given to the computer configuration, the available resources, and the nature of the processors involved. This section contains some ideas and suggestions which may be of assistance or of general relevance.

5.1 Table B representation and access

If table B were completely filled with entries, it would contain 64 x 256 x 5 = 81,920 items of information, which could be accommodated in just over 1 megabyte of computer memory. All references would be direct and extremely efficient (see Fig. 5.1).

Most installations will not wish to use such a large amount of memory. Table B will always be sparsely filled. The current specification defines less than 5% of the possible space. Thus a memory resident table B could be accommodated using about 60 Kilobytes, but would require a means of indirect addressing. This may be achieved by using an index based on classes, pointing to the first entry in each class. From this entry a short search would be required to locate the required entry (see Fig. 5.2).

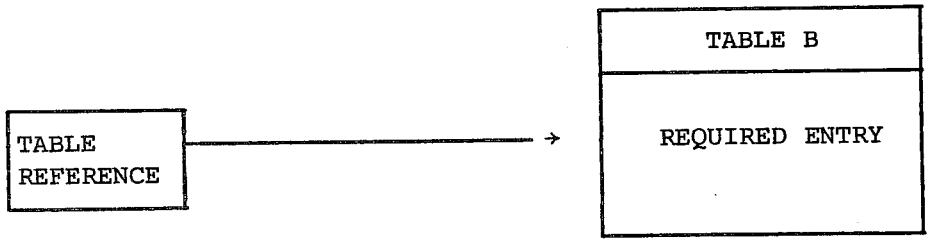

Fig. 5.1: Direct Addressing

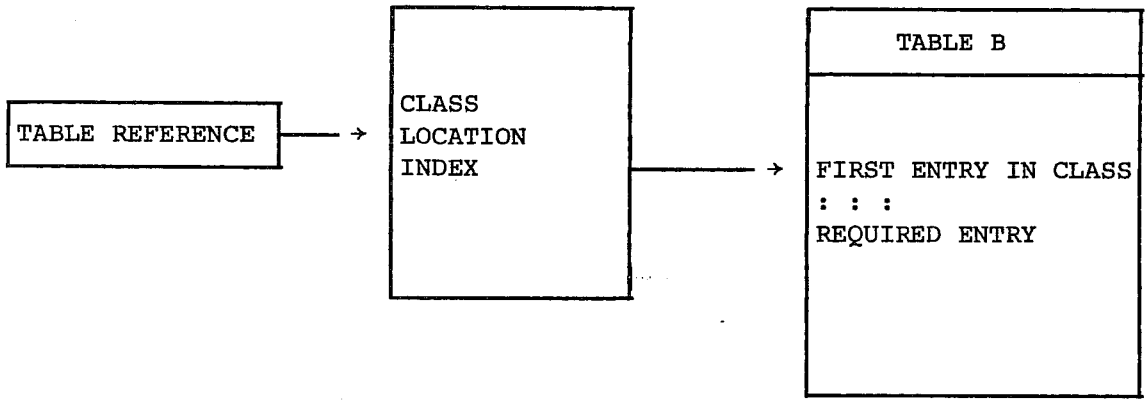

Fig. 5.2: Indirect Addressing with Final Short Search

In small installations where memory resources are insufficient to retain the whole table in memory, each class could constitute a separate record on a direct access data set. The class would then indicate the record to be accessed. In such circumstances it might be worth adding an indicator to each working table entry. Entries corresponding to standard table B entries could then be tagged, and searched before initiating the read of a table B record. Such entries could then be re-used to avoid unnecessary input/output.

5.2 Table D representation and access

For processing purposes, a table D entry must contain a list of descriptors and the length of that list. For indirect addressing the table reference value is also required. There is no fixed maximum list length, but a parameterised value set to about 20 should be sufficient.

Table D will always be sparsely filled, thus in direct addressing using an index of categories, and location techniques similar to those suggested for table B would seem appropriate. Where input/output techniques are necessary to reduce memory requirements, each category could constitute a direct access data record.

5.3 Working tables

Working tables will be required to contain the working table for data identification and the table containing augmented table B values. The contents of these working tables are defined fully in section 4 above.

6. EXAMPLE OF DECODING BUFR MESSAGES

A single complete BUFR entity is a string of binary numbers. Therefore it is not suitable for human interpretations without computers. To visualise BUFR messages in these examples the notation $(value)_N$ is used to indicate the representation of a binary value in N bits of the BUFR message. The binary values are written as decimal numbers for convenience; CCITT IA5 character values are written as characters.

6.1 BUFR messages containing SURFACE reports

(BUFR)$_{32}$			
(18)$_{24}$	(0)$_{8}$	(98)$_{16}$	(0)$_{8}$
(0)$_{8}$	(0)$_{8}$	(0)$_{8}$	(0)$_{16}$
(87)$_{8}$	(5)$_{8}$	(23)$_{8}$	(12)$_{8}$
(0)$_{8}$	(0)$_{8}$		
(10)$_{24}$	(0)$_{8}$	(1)$_{16}$	(1)$_{8}$
(307003)$_{16}$	(0)$_{8}$		
(48)$_{24}$	(0)$_{8}$	(13)$_{7}$	(275)$_{10}$
(1)$_{2}$	(1987)$_{12}$	(5)$_{4}$	(23)$_{6}$
(12)$_{5}$	(0)$_{6}$	(13448000)$_{25}$	(20020000)$_{26}$
(538)$_{15}$	(9956)$_{14}$	(9993)$_{14}$	(517)$_{10}$
(7)$_{4}$	(130)$_{9}$	(45)$_{12}$	(2956)$_{12}$
(2937)$_{12}$	(92)$_{7}$	(300)$_{13}$	(81)$_{8}$
(6)$_{4}$	(15)$_{4}$	(80)$_{7}$	(1)$_{6}$
(6)$_{4}$	(120)$_{11}$	(36)$_{6}$	(22)$_{6}$
(63)$_{6}$	(2)$_{8}$	(1)$_{6}$	(6)$_{4}$
(42)$_{6}$	(160)$_{11}$	(2)$_{6}$	(2)$_{4}$
(27)$_{6}$	(220)$_{11}$	(0)$_{5}$	(7777)$_{32}$

Fig. 6.1: SURFACE message in BUFR

Fig. 6.1 illustrates the contents of a BUFR message containing a single SURFACE report. The steps required to decode this message are indicated below.

Step 1: Expansion of the identification section gives the array ISEC1. The values obtained are as follows:

SUBSCRIPT	CONTENTS
1	18
2	0
3	98
4 to 8	0
9	87
10	5
11	23
12	12
13 to 14	0

The example contains 18 octets, representing 14 data items, thus IDIM1 = 14. (cf section 3.2 above).

Step 2: Expansion of the optional section is trivial. Since ISEC1(5) is zero, this section is not included; IDIM2 = 1, and ISEC2 contains only 1 entry, i.e. ISEC2(1) = 0 (cf section 3.3 above).

Step 3: Expansion of the preliminary information of the data description section gives the array ISEC3. The values obtained are as follows:

SUBSCRIPT	CONTENTS
1	10
2	0
3	1
4	1

Note that the number of data subsets = N = ISEC3(3) = 1.
There are by definition 7 octets containing 4 preliminary data items, and IDIM3 = 4. Since ISEC3 (1) = 10, the number of octets available to represent descriptors is (10-7) = 3. Each descriptor requires 2 octets, thus there exists 1 descriptor (cf section 3.4 above).

Step 4: Expansion of the data description is performed using the processes described in section 4.2 above:

- the single descriptor 307003 is replaced by the list 307001, 101000, 031001, 302005 from table D;
- 307001 is replaced by the list 301031, 302011;
- 301031 is replaced by the list 301001, 002001, 301011, 301012, 301022;
- 301011 is replaced by the list 001001, 001002;
- 001001 is an element descriptor, so the appropriate table B entry becomes the first entry in the working table;
- 001002 is an element descriptor, so the appropriate table B entry becomes the second entry in the working table;
- 002001 is an element descriptor, so the appropriate table B entry becomes the third entry in the working table;
- 301011 is a sequence descriptor, so is replaced by the list 004001, 004002, 004003;
- 004001 is an element descriptor, so the appropriate table B entry becomes the fourth entry in the working table;

- - -

- 101000 indicates delayed replication of 1 descriptor; set K = 1 (see Fig. 4.6);

- 031001 is an element descriptor, indicating a replication factor within the data; the corresponding table B entry is added to the working table;

- using the working table so far constructed, the replication factor is located and extracted from the data; this value (2) is assigned to J;

- the 1 descriptor to be replicated, 302005, is replaced by 2 copies,completing the replication operation;

- 302005 is a sequence descriptor, so is replaced by the list 008002, 020011, 020012, 020013 from table D;

- 008002 is an element descriptor, so the corresponding table B entry is added to the working table;

\- - -

Although not all steps are shown, it will be seen from the above selection of steps that the following working table results:

W O R K I N G T A B L E

Table reference	Element name	Units	Scale	Ref. value	Data width
001001	WMO Block Number				7
001002	WMO Station number				10
002001	Type of station	Code table 002001			2
004001	Year	Year	0	0	12
004002	Month	Month	0	0	4
004003	Day	Day	0	0	6
004004	Hour	Hour	0	0	5
004005	Minute	Minute	0	0	6
005001	Latitude (high accuracy)	Degrees	5	-9000000	25
006001	Longitude (high accuracy)	Degrees	5	-18000000	26
007001	Height of station	m	0	-400	15
010004	Pressure	Pa	-1	0	14
010051	Pressure reduced to msl	Pa	-1	0	14
010061	3 hour pressure change	Pa	-1	-500	10
010063	Characteristic of Pressure tendency	Code table 010063			4
011011	Wind direction at 10m	Degrees true	0	0	9
011012	Wind speed at 10m	m/s	1	0	12
012004	Dry bulb temperature at 2m	K	1	0	12
012006	Dew point temperature at 2m	K	1	0	12
013003	Relative humidity	%	0	0	7
020001	Horizontal visibility	m	-1	0	13
020003	Present weather	Code table 020003		0	8
020004	Past weather (1)	Code table 020003		0	4
020005	Past weather (2)	Code table 020005		0	4

Table reference	Element name	Units	Scale	Ref. value	Data width
020010	Cloud cover (total)	%		0	7
008002	Vertical Significance (surface observations)	Code table 008002			6
020011	Cloud amount	Code table 020011		0	4
020013	Height of base of cloud	m	-1	-40	11
020012	Cloud type	Code table 020012		0	6
020012	Cloud type	Code table 020012		0	6
020012	Cloud type	Code table 020012		0	6
031001	Delayed replication factor		0	0	8
008002	vertical significance (surface observations)	Code table 008002			6
020011	Cloud amount	Code table 020011		0	4
020012	Cloud type	Code table 020012		0	6
020013	Height of base of cloud	m	-1	-40	11
008002	vertical significance (surface observations)	Code table 008002			6
020011	Cloud amount	Code table 020011		0	4
020012	Cloud type	Code table 020012		0	6
020013	Height of base of cloud	m	-1	-40	11

Subscript	CNAMES	CUNITS
1	WMO Block number	
2	WMO Station number	
3	Type of station	Code table 002001
4	Year	Year
5	Month	Month
6	Day	Day
7	Hour	Hour
8	Minute	Minute
9	Latitude (high Accuracy)	degrees
10	Longitude (high accuracy)	degrees
11	Height of station	m
12	Pressure	Pa
13	Pressure reduced to MSL	Pa
14	3 hour pressure change	Pa
15	characteristic of pressure tendency	Code table 010063
16	Wind direction at 10m	degrees true
17	Wind speed at 10m	m/s
18	Dry bulb temperature at 2m	K
19	Dew point temperature at 2m	K
20	Relative humidity	%
21	Horizontal visibility	m
22	Present weather	Code table 020003
23	Past weather (1)	Code table 020004
24	Past weather (2)	Code table 020005
25	Cloud cover (total)	%
26	Vertical significance (surface observations)	Code table 008002
27	Cloud amount	Code table 020011
28	Height of base of cloud	m
29	Cloud type	Code table 020012
30	Cloud type	Code table 020012
31	Cloud type	Code table 020012

32	Vertical significance (surface observations)	Code table 008002
33	Cloud amount	Code table 020011
34	Cloud type	Code table 020012
35	Height of base of cloud	m
36	Vertical significance (surface observations)	Code table 008002
37	Cloud amount	Code table 020011
38	Cloud type	Code table 020012
39	Height of base of cloud	m

step 5 Expansion of the preliminary items of the data section gives the array ISEC4, which has a dimension IDIM4=2, as follows:

Subscript	Contents
1	48
2	0

(cf section 3.5 above)

step 6 Using the working table, and following figure 4.9, the data are decoded into array VALUES giving:-

Subscript	Contents	Subscript	Contents
(1,1)	13.	(21,1)	3000.
(2,1)	275.	(22,1)	81.
(3,1)	1.	(23,1)	6.
(4,1)	1987.	(24,1)	15.
(5,1)	5.	(25,1)	80.
(6,1)	23.	(26,1)	1.
(7,1)	12.	(27,1)	6.
(8,1)	0.	(28,1)	800.
(9,1)	44.8	(29,1)	36.
(10,1)	20.2	(30,1)	22.
(11,1)	138.	(31,1)	63.
(12,1)	995.6	(32,1)	1.
(13,1)	999.3	(33,1)	6.
(14,1)	1.7	(34,1)	42.
(15,1)	7.	(35,1)	1200.
(16,1)	130.	(36,1)	2.
(17,1)	4.5	(37,1)	2.
(18,1)	295.6	(38,1)	27.
(19,1)	293.7	(39,1)	1800.
(20,1)	92.		

No character data were encountered, thus CVALS is returned empty.

step 7 Gathering together the supplementary information into array ISUP gives:

Subscript	Contents	
1	14	(IDIM1)
2	1	(IDIM2)
3	4	(IDIM3)
4	39	(M)
5	1	(N)
6	0	(IDIMC)
7	84	(message length)
8-16	0	

6.2 SURFACE report expanded to display form

To produce the display form described in section 3.6 above requires only that those data values representing code figures or flag values be replaced by character data from the code tables or flag tables respectively. Thus, the only returned values that woyuld be changed would be VALUES and IDIMC CVALS. Completing this expansion gives the following entries in VALUES:

Subscript	Contents	Subscript	Contents
(1,1)	13.	(21,1)	3000.
(2,1)	275.	(22,1)	3032.
(3,1)	1014.	(23,1)	4004.
(4,1)	1987.	(24,1)	5013.
(5,1)	5.	(25,1)	80.
(6,1)	23.	(26,1)	6029.
(7,1)	12.	(27,1)	7007.
(8,1)	0.	(28,1)	800.
(9,1)	44.8	(29,1)	8027.
(10,1)	20.2	(30,1)	9047.
(11,1)	138.	(31,1)	10013.
(12,1)	995.6	(32,1)	11029.
(13,1)	999.3	(33,1)	12007.
(14,1)	1.7	(34,1)	13013.
(15,1)	2032.	(35,1)	1200.
(16,1)	130.	(36,1)	14029.
(17,1)	4.5	(37,1)	15007.
(18,1)	295.6	(38,1)	16011.
(19,1)	293.7	(39,1)	18000.

16 values refer to code tables; thus CVALS contains 16 entries as follows:

Subscript	Contents
1	(manned station)$_{14}$
2	(decreasing (steadily or unsteadily))$_{32}$
3	(Rain showers, moderate or heavy)$_{32}$
4	(Rain)$_{4}$
5	(Missing value)$_{13}$
6	(First significant cloud layer)$_{29}$
7	(55%-64%)$_{7}$
8	(Stratocumulus cumulogenitus)$_{27}$
9	(Altocumulus cumulogenitus (or cumulonimbogenitus))$_{47}$
10	(Missing value)$_{13}$
11	(First significant cloud layer)$_{29}$
12	(55%-64%)$_{7}$
13	(Stratocumulus)$_{13}$
14	(second significant cloud layer)$_{29}$
15	(15%-24%)$_{7}$
16	(Altostratus)$_{11}$

Finally, the supplementary information, ISUP, will contain:

Subscript	Contents
1	14 (IDIM1)
2	1 (IDIM2)
3	4 (IDIM3)
4	39 (M)
5	1 (N)
6	16 (IDIMC)
7	84 (message length)
8-16	9

Note that with this detail of information returned it is trivial to produce a display application enabling the data to be displayed or printed in tabular form. The resulting display contains no unresolved references to tables, and thus contains all required information. It can be readily understood by users lacking familiarity with the code tables and code forms.

6.3 BUFR message containing compressed satellite data

In the following example, the results of decoding a BUFR message containing 3 satellite soundings in compressed form are presented. Figure 6.2 illustrates the BUFR message before decoding.

```
(BUFR)32,
(     18 )24,   (      0 )8,    (     98 )16,   (      0 )8,
(      0 )8,    (      3 )8,    (      0 )8,    (      0 )16,
(     87 )8,    (      5 )8,    (     14 )8,    (     21 )8,
(     30 )8,    (      0 )8,
(     10 )24,   (      0 )8,    (      3 )16,   (      3 )8,
( 301002 )16,   (      0 )8,
(    200 )24,   (      0 )8,    (     33 )10,   (      0 )6,
(      1 )9,    (      0 )6,    (      4 )8,    (      0 )6,
(   1987 )12,   (      0 )6,    (      5 )4,    (      0 )6,
(     14 )6,    (      0 )6,    (     21 )5,    (      0 )6,
(     30 )6,    (      2 )6,    (      0 )2,    (      1 )2,
(      2 )2,    (12560000)25,   (     19 )6,    (      0 )19,
( 300000 )19,   (400000  )19,   (27300000)26,   (      0 )6,
(  10000 )14,   (      5 )6,    (      0 )5,    (     10 )5,
(     30 )5,    (      0 )6,    (      0 )6,    ( 16370  )15,
(      8 )6,    (    160 )8,    (     60 )8,    (      0 )8,
(  14900 )15,   (      7 )6,    (    100 )7,    (      0 )7,
(      0 )7,    (      0 )2,    (      1 )6,    (      0 )1,
(      1 )1,    (      1 )1,    (   2951 )12,   (      4 )6,
(     10 )4,    (      2 )4,    (      0 )4,    (      2 )4,
(      2 )6,    (      0 )2,    (      2 )2,    (      1 )2,
(   4000 )14,   (     10 )6,    (      0 )10,   (      0 )10,
(    300 )10,   (  10000 )14,   (      5 )6,    (      0 )5,
(     10 )5,    (     30 )5,    (   8500 )14,   (      0 )6,
```

```
(      7 )6,    (      0 )6,    (     80 )7,    (      3 )6,
(      5 )3,    (      5 )3,    (      0 )3,    (   2811 )12,
(      5 )6,    (     18 )5,    (     16 )5,    (      0 )5,
(   8500 )14,   (      0 )6,    (   7000 )14,   (      0 )6,
(      7 )6,    (      0 )6,    (     90 )7,    (      3 )6,
(      0 )3,    (      5 )3,    (      0 )3,    (   2731 )12,
(      6 )6,    (     33 )6,    (     10 )6,    (      0 )6,
(   7000 )14,   (      0 )6,    (   5000 )14,   (      0 )6,
(      7 )6,    (      0 )6,    (     20 )7,    (      7 )6,
(      0 )7,    (     75 )7,    (     60 )7,    (   2611 )12,
(      5 )6,    (      0 )5,    (     10 )5,    (     20 )5,
(   5000 )14,   (      0 )6,    (   4000 )14,   (      0 )6,
(      7 )6,    (      0 )6,    (     80 )7,    (      0 )6,
(   2461 )12,   (      5 )6,    (     10 )5,    (      0 )5,
(     30 )5,,   (   4000 )14,   (      0 )6,    (   3000 )14,
(      0 )6,    (      7 )6,    (      0 )6,    (     80 )7,
(      3 )6,    (      0 )3,    (      5 )3,    (      5 )3,
(   2354 )12,   (      5 )6,    (      0 )5,    (      7 )5,
(     17 )5,    (   3000 )14,   (      0 )6,    (   2000 )14,
(      0 )6,    (      7 )6,    (      0 )6,    (     80 )7,
(      3 )6,    (      5 )3,    (      0 )3,    (      5 )3,
(   2204 )12,   (      3 )6,    (      7 )3,    (      0 )3,
(      6 )3,    (   2000 )14,   (      0 )6,    (   1000 )14,
(      0 )6,    (      7 )6,    (      0 )6,    (     70 )7,
(      5 )6,    (     10 )5,    (     20 )5,    (      0 )5,
(   2166 )12,   (      4 )6,    (      5 )4,    (     10 )4,
(      0 )4,    (   1000 )14,   (      0 )6,    (    700 )14,
(      0 )6,    (      7 )6,    (      0 )6,    (     50 )7,
(      6 )6,    (      0 )6,    (     30 )6,    (     40 )6,
(   2168 )12,   (      2 )6,    (      2 )2,    (      0 )2,
(      3 )2,    (    700 )14,   (      0 )6,    (    500 )14,
(      0 )6,    (      7 )6,    (      0 )6,    (     80 )7,
(      3 )6,    (      1 )3,    (      4 )3,    (      0 )3,
(   2167 )12,   (      3 )6,    (      2 )3,    (      4 )3,
(      0 )3,    (      0 )10,   (   7777 )32.
```

Figure 6.2 BUFR satellite message

During decoding the following working table is generated:

W O R K I N G T A B L E

Table reference	Element name	Unit	Scale	Ref. value	Data width	IW	IWS
001007	Satellite identifier	Code table 001007			10	0	16
002021	Satellite instrument data used in processing	Flag table 002021			9	0	15
002022	Satellite data processing technique	Code table 002022			8	0	14
004001	Year	Year	0	0	12	0	18
004002	Month	Month	0	0	4	0	10
004003	Day	Day	0	0	6	0	12
004004	Hour	Hour	0	0	5	0	11
004005	Minute	Minute	0	0	6	2	18
005001	Latitude (high accuracy)	Degrees	5	-9000000	25	19	88
006001	Longitude (high accuracy)	Degrees	5	-18000000	26	0	32
007004	Pressure	Pa	-1	0	14	5	35
008003	Vertical significance (satellite observations)	Code table 008003			6	0	12
007021	Elevation	Degrees	2	-9000	15	8	45
007022	Solar elevation	Degrees	2	-9000	15	7	42
008012	Land/sea qualifier	Code table 008012			2	1	11
012061	Skin temperature	K	1	0	12	4	30
020011	Cloud amount	Code table		0	4	2	16
020016	Pressure at the top of cloud	Pa	-1	0	14	10	50
007004	Pressure	Pa	-1	0	14	5	35
007004	Pressure	Pa	-1	0	14	0	20
031002	Meaning of additional field	Code table 031002			6	0	12
	Associated field				7	3	22

Table reference	Element name	Unit	Scale	Ref. value	Data width	IW	IWS
012001	Temperature	K	1	0	12	5	33
007004	Pressure	Pa	-1	0	14	0	20
007004	Pressure	Pa	-1	0	14	0	20
031002	Meaning of additional field	Code table 031002			6	0	12
	Associated field				7	3	22
012001	Temperature	K	1	0	12	6	36
007004	Pressure	Pa	-1	0	14	0	20
007004	Pressure	Pa	-1	0	14	0	20
031002	Meaning of additional field	Code table 031002			6	0	12
	Associated field				7	7	34
012001	Temperature	K	1	0	12	5	33
007004	Pressure	Pa	-1	0	14	0	20
007004	Pressure	Pa	-1	0	14	0	20
031002	Meaning of additional field	Code table 031002			6	0	12
	Associated field				7	0	13
012001	Temperature	K	1	0	12	5	33
007004	Pressure	Pa	-1	0	14	0	20
007004	Pressure	Pa	-1	0	14	0	20
031002	Meaning of additional field	Code table 031002			6	0	12
	Associated field				7	3	22
012001	Temperature	K	1	0	12	5	33
007004	Pressure	Pa	-1	0	14	0	20
007004	Pressure	Pa	-1	0	14	0	20

Table reference	Element name	Unit	Scale	Ref. value	Data width	IW	IWS
031002	Meaning of additional field	Code table 031002			6	0	12
	Associated field				7	3	22
012001	Temperature	K	1	0	12	3	27
007004	Pressure	Pa	-1	0	14	0	20
007004	Pressure	Pa	-1	0	14	0	20
031002	Meaning of additional field	Code table 031002			6	0	12
	Associated field				7	5	28
012001	Temperature	K	1	0	12	4	30
007004	Pressure	Pa	-1	0	14	0	20
007004	Pressure	Pa	-1	0	14	0	20
031002	Meaning of additional field	Code table 031002			6	0	12
	Associated field				7	6	31
012001	Temperature	K	1	0	12	2	24
007004	Pressure	Pa	-1	0	14	0	20
007004	Pressure	Pa	-1	0	14	0	20
031002	Meaning of additional field	Code table 031002			6	0	12
	Associated field				7	3	22
042001	Temperature	K	1	0	12	3	27

Using this working table to decode the data, the following values are returned:

Array ISEC1 (IDIM1=14)

Subscript	Contents
1	18
2	0
3	98
4	0
5	0
6	3
7	0

8	0
9	87
10	5
11	14
12	21
13	30
14	0
15 to 18	0

Array ISEC2 (IDIM2=1)

Subscript	Contents
1	0

Array ISEC3 (IDIM3=4)

Subscript	Contents
1	10
2	0
3	3
4	3

Arrays CNAMES and CUNITS (M=63)

Subscript	CNAMES	CUNITS
1	SATELLITE IDENTIFIER	CODE TABLE 001007
2	SATELLITE INSTRUMENT DATA USED IN PROCESSING	FLAG TABLE 002021
3	SATELLITE DATA PROCESSING TECHNIQUE USED	CODE TABLE 002022
4	YEAR	YEAR
5	MONTH	MONTH
6	DAY	DAY
7	HOUR	HOUR
8	MINUTE	MINUTE
9	LATITUDE (HIGH ACCURACY)	DEGREES

Subscript	CNAMES	CUNITS
10	LONGITUDE (HIGH ACCURACY)	DEGREES
11	PRESSURE	PA
12	VERTICAL SIGNIFICANCE (SATELLITE OBSERVATIONS)	CODE TABLE 008003
13	ELEVATION	DEGREES
14	SOLAR ELEVATION	DEGREES
15	LAND/SEA QUALIFIER	CODE TABLE 008012
16	SKIN TEMPERATURE	K
17	CLOUD AMOUNT	CODE TABLE 020011
18	PRESSURE AT THE TOP OF CLOUD	PA
19	PRESSURE	PA
20	PRESSURE	PA
21	MEANING OF ADDITIONAL FIELD	CODE TABLE 031002
22	ASSOCIATED FIELD	
23	TEMPERATURE	K
24	PRESSURE	PA
25	PRESSURE	PA
26	MEANING OF ADDITIONAL FIELD	CODE TABLE 031002
27	ASSOCIATED FIELD	
28	TEMPERATURE	K
29	PRESSURE	PA
30	PRESSURE	PA
31	MEANING OF ADDITIONAL FIELD	CODE TABLE 031002
32	ASSOCIATED FIELD	
33	TEMPERATURE	K
34	PRESSURE	PA
35	PRESSURE	PA
36	MEANING OF ADDITIONAL FIELD	CODE TABLE 031002
37	ASSOCIATED FIELD	
38	TEMPERATURE	K
39	PRESSURE	PA
40	PRESSURE	PA

Subscript	CNAMES	CUNITS
41	MEANING OF ADDITIONAL FIELD	CODE TABLE 031002
42	ASSOCIATED FIELD	
43	TEMPERATURE	K
44	PRESSURE	PA
45	PRESSURE	PA
46	MEANING OF ADDITIONAL FIELD	CODE TABLE 031002
47	ASSOCIATED FIELD	
48	TEMPERATURE	K
49	PRESSURE	PA
50	PRESSURE	PA
51	MEANING OF ADDITIONAL FIELD	CODE TABLE 031002
52	ASSOCIATED FIELD	
53	TEMPERATURE	K
54	PRESSURE	PA
55	PRESSURE	PA
56	MEANING OF ADDITIONAL FIELD	CODE TABLE 031002
57	ASSOCIATED FIELD	
58	TEMPERATURE	K
59	PRESSURE	PA
60	PRESSURE	PA
61	MEANING OF ADDITIONAL FIELD	CODE TABLE 031002
62	ASSOCIATED FIELD	
63	TEMPERATURE	K

Array ISEC4 (IDIM=2)

Subscript	Contents
1	200
2	0

Array VALUES (M=63, N=3)

First subscript	N=1	N=2	N=3
1	33.	33.	33.
2	0.	0.	0.
3	1.	1.	1.
4	1987.	1987.	1987.
5	5.	5.	5.
6	14.	14.	14.
7	21.	21.	21.
8	30.	31.	32.
9	35.6	38.6	39.6
10	93.0	93.0	93.0
11	1000.0	1001.0	1003.0
12	0.	0.	0.
13	75.3	74.3	73.7
14	60.0	59.0	59.0
15	0.	1.	1.
16	296.1	295.3	295.1
17	2.	4.	1.
18	40000.0	50000.0	43000.0
19	10000.0	10010.0	10030.0
20	85000.0	85000.0	85000.0
21	7.	7.	7.
22	85.	85.	80.
23	282.9	282.7	281.1
24	85000.	85000.	85000.
25	70000.	70000.	70000.
26	7.	7.	7.
27	90.	95.	90.
28	276.4	274.1	273.1
29	70000.	20000.	70000.

30	50000.	50000.	50000.
31	7.	7.	7.
32	20.	95.	80.
33	261.1	262.1	263.1
34	50000.	50000.	50000.
35	40000.	40000.	40000.
36	7.	7.	7.
37	80.	80.	80.
38	247.1	246.1	249.1
39	40000.	40000.	40000.
40	30000.	30000.	30000.
41	7.	7.	7.
42	80.	85.	85.
43	235.4	236.1	237.1
44	30000.	30000.	30000.
45	20000.	20000.	20000.
46	7.	7.	7.
47	85.	80.	85.
48	221.1	220.4	221.0
49	20000.	20000.	20000.
50	10000.	10000.	10000.
51	7.	7.	7.
52	80.	90.	70.
53	217.1	217.6	216.6
54	10000.	10000.	10000.
55	7000.	7000.	7000.
56	7.	7.	7.
57	50.	80.	90.
58	217.0	216.8	217.1
59	7000.	7000.	5000.
60	5000.	5000.	5000.
61	7.	7.	7.
62	81.	84.	80.
63	216.9	217.1	216.7

Array CVALS (IBIMC=0)

No data is returned in CVALS, as no character data were encountered.

Array ISUP (16 items)

Subscript	Contents
1	14 (IDIM1)
2	1 (IDIM2)
3	4 (IDIM3)
4	2 (IDIM4)
5	63 (M)
6	3 (N)
7	0 (IDIMC)
8	242 (message length)
9-16	0

7. CONCLUDING REMARKS

The value of a standard interface to decoded FM94 BUFR messages lies within the resulting standardisation and transportability of subsequent applications. The interface proposed above contains all of the relevant data, related to high level computer languages rather than particular machine representations, and is straight forward to apply. The extension to the "display form" enables BUFR data to be displayed or printed without the user needing to refer to code tables. Both forms contain an array of names, CNAMES, which can be used after decoding to locate individual elements.

The decoding strategy proposed enables an expansion of the full BUFR message, or of selected elements. The concept of the working table enables the binary data to be "navigated" - any required item can be located, expanded and decoded without the need to decode the complete data section.

The examples in section 6 were worked by hand, following strictly the figures and algorithms contained in sections 3 and 4. This was found to be a useful way of verifying both the logic presented and the feasibility of FM94 BUFR as currently defined.