INTERNATIONAL TELECOMMUNICATION UNION

CCITT

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

BLUE BOOK

VOLUME X – FASCICLE X.7

MAN-MACHINE LANGUAGE (MML)

RECOMMENDATIONS Z.301-Z.341

IXTH PLENARY ASSEMBLY
MELBOURNE, 14-25 NOVEMBER 1988

Geneva 1989

INTERNATIONAL TELECOMMUNICATION UNION

CCITT

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

BLUE BOOK

VOLUME X – FASCICLE X.7

MAN-MACHINE LANGUAGE (MML)

RECOMMENDATIONS Z.301-Z.341

IXTH PLENARY ASSEMBLY
MELBOURNE, 14-25 NOVEMBER 1988

Geneva 1989

ISBN 92-61-03811-5

Printed in Switzerland

CONTENTS OF THE CCITT BOOK
APPLICABLE AFTER THE NINTH PLENARY ASSEMBLY (1988)

BLUE BOOK

FASCICLE III.6 – Line transmission of non-telephone signals. Transmission of sound-programme and television signals. Series H and J Recommendations (Study Group XV).

FASCICLE III.7 – Integrated Services Digital Network (ISDN) – General structure and service capabilities. Recommendations I.110-I.257 (Study Group XVIII).

FASCICLE III.8 – Integrated Services Digital Network (ISDN) – Overall network aspects and functions, ISDN user-network interfaces. Recommendations I.310-I.470 (Study Group XVIII).

FASCICLE III.9 – Integrated Services Digital Network (ISDN) – Internetwork interfaces and maintenance principles. Recommendations I.500-I.605 (Study Group XVIII).

Volume IV

FASCICLE IV.1 – General maintenance principles: maintenance of international transmission systems and telephone circuits. Recommendations M.10-M.782 (Study Group IV).

FASCICLE IV.2 – Maintenance of international telegraph, phototelegraph and leased circuits. Maintenance of the international public telephone network. Maintenance of maritime satellite and data transmission systems. Recommendations M.800-M.1375 (Study Group IV).

FASCICLE IV.3 – Maintenance of international sound-programme and television transmission circuits. Series N Recommendations (Study Group IV).

FASCICLE IV.4 – Specifications for measuring equipment. Series O Recommendations (Study Group IV).

Volume V – Telephone transmission quality. Series P Recommendations (Study Group XII).

Volume VI

FASCICLE VI.1 – General Recommendations on telephone switching and signalling. Functions and information flows for services in the ISDN. Supplements. Recommendations Q.1-Q.118 *bis* (Study Group XI).

FASCICLE VI.2 – Specifications of Signalling Systems Nos. 4 and 5. Recommendations Q.120-Q.180 (Study Group XI).

FASCICLE VI.3 – Specifications of Signalling System No. 6. Recommendations Q.251-Q.300 (Study Group XI).

FASCICLE VI.4 – Specifications of Signalling Systems R1 and R2. Recommendations Q.310-Q.490 (Study Group XI).

FASCICLE VI.5 – Digital local, transit, combined and international exchanges in integrated digital networks and mixed analogue-digital networks. Supplements. Recommendations Q.500-Q.554 (Study Group XI).

FASCICLE VI.6 – Interworking of signalling systems. Recommendations Q.601-Q.699 (Study Group XI).

FASCICLE VI.7 – Specifications of Signalling System No. 7. Recommendations Q.700-Q.716 (Study Group XI).

FASCICLE VI.8 – Specifications of Signalling System No. 7. Recommendations Q.721-Q.766 (Study Group XI).

FASCICLE VI.9 – Specifications of Signalling System No. 7. Recommendations Q.771-Q.795 (Study Group XI).

FASCICLE VI.10 – Digital subscriber signalling system No. 1 (DSS 1), data link layer. Recommendations Q.920-Q.921 (Study Group XI).

FASCICLE VI.11 – Digital subscriber signalling system No. 1 (DSS 1), network layer, user-network management. Recommendations Q.930-Q.940 (Study Group XI).

FASCICLE VI.12 – Public land mobile network. Interworking with ISDN and PSTN. Recommendations Q.1000-Q.1032 (Study Group XI).

FASCICLE VI.13 – Public land mobile network. Mobile application part and interfaces. Recommendations Q.1051-Q.1063 (Study Group XI).

FASCICLE VI.14 – Interworking with satellite mobile systems. Recommendations Q.1100-Q.1152 (Study Group XI).

Volume VII

FASCICLE VII.1 – Telegraph transmission. Series R Recommendations. Telegraph services terminal equipment. Series S Recommendations (Study Group IX).

FASCICLE VII.2 – Telegraph switching. Series U Recommendations (Study Group IX).

FASCICLE VII.3 – Terminal equipment and protocols for telematic services. Recommendations T.0-T.63 (Study Group VIII).

FASCICLE VII.4 – Conformance testing procedures for the Teletex Recommendations. Recommendation T.64 (Study Group VIII).

FASCICLE VII.5 – Terminal equipment and protocols for telematic services. Recommendations T.65-T.101, T.150-T.390 (Study Group VIII).

FASCICLE VII.6 – Terminal equipment and protocols for telematic services. Recommendations T.400-T.418 (Study Group VIII).

FASCICLE VII.7 – Terminal equipment and protocols for telematic services. Recommendations T.431-T.564 (Study Group VIII).

Volume VIII

FASCICLE VIII.1 – Data communication over the telephone network. Series V Recommendations (Study Group XVII).

FASCICLE VIII.2 – Data communication networks: services and facilities, interfaces. Recommendations X.1-X.32 (Study Group VII).

FASCICLE VIII.3 – Data communication networks: transmission, signalling and switching, network aspects, maintenance and administrative arrangements. Recommendations X.40-X.181 (Study Group VII).

FASCICLE VIII.4 – Data communication networks: Open Systems Interconnection (OSI) – Model and notation, service definition. Recommendations X.200-X.219 (Study Group VII).

FASCICLE VIII.5 – Data communication networks: Open Systems Interconnection (OSI) – Protocol specifications, conformance testing. Recommendations X.220-X.290 (Study Group VII).

FASCICLE VIII.6 – Data communication networks: interworking between networks, mobile data transmission systems, internetwork management. Recommendations X.300-X.370 (Study Group VII).

FASCICLE VIII.7 – Data communication networks: message handling systems. Recommendations X.400-X.420 (Study Group VII).

FASCICLE VIII.8 – Data communication networks: directory. Recommendations X.500-X.521 (Study Group VII).

Volume IX – Protection against interference. Series K Recommendations (Study Group V). Construction, installation and protection of cable and other elements of outside plant. Series L Recommendations (Study Group VI).

Volume X

FASCICLE X.1 – Functional Specification and Description Language (SDL). Criteria for using Formal Description Techniques (FDTs). Recommendation Z.100 and Annexes A, B, C and E, Recommendation Z.110 (Study Group X).

FASCICLE X.2 – Annex D to Recommendation Z.100: SDL user guidelines (Study Group X).

FASCICLE X.3 – Annex F.1 to Recommendation Z.100: SDL formal definition. Introduction (Study Group X).

FASCICLE X.4 – Annex F.2 to Recommendation Z.100: SDL formal definition. Static semantics (Study Group X).

FASCICLE X.5 – Annex F.3 to Recommendation Z.100: SDL formal definition. Dynamic semantics (Study Group X).

FASCICLE X.6 – CCITT High Level Language (CHILL). Recommendation Z.200 (Study Group X).

FASCICLE X.7 – Man-Machine Language (MML). Recommendations Z.301-Z.341 (Study Group X).

CONTENTS OF FASCICLE X.7 OF THE BLUE BOOK

Recommendations Z.301 to Z.341

Man-machine language (MML)

PRELIMINARY NOTES

1 The Questions entrusted to each Study Group for the Study Period 1989-1992 can be found in Contribution No. 1 to that Study Group.

2 In this Fascicle, the expression "Administration" is used for shortness to indicate both a telecommunication Administration and a recognized private operating agency.

FASCICLE X.7

Recommendations Z.301 to Z.341

MAN-MACHINE LANGUAGE (MML)

PAGE INTENTIONALLY LEFT BLANK

PAGE LAISSEE EN BLANC INTENTIONNELLEMENT

SECTION 1

GENERAL PRINCIPLES

Recommendation Z.301

INTRODUCTION TO THE CCITT MAN-MACHINE LANGUAGE

1 Field of application

The man-machine language (CCITT MML) can be used to facilitate operation and maintenance functions of Stored Program Control SPC systems of different types. Depending upon national requirements, CCITT MML can also be used to facilitate installation and acceptance testing of such systems.

In many cases, SPC systems will be supported by auxiliary systems, e.g., in operation and maintenance centres and/or centres for other purposes such as sales, subscribers' complaints, etc., to carry out functions in cooperation with the SPC system. Different types of communication may be required for this cooperation. To clarify where the CCITT MML is intended to be used, a configuration is shown in Figure 1/Z.301 which illustrates the case of three separate systems. Local and remote man-machine terminals may be used. The configuration of systems in a network may vary, but this does not alter the principles governing the field of application of the MML.

The CCITT MML is intended to handle the functions required at the interface marked 1 while other methods may be required for the interface marked 2. Interface 2 is not considered. Since interface 1 is the interface of interest, it should be stressed that no assumptions are made concerning the physical location of any supporting software or whether, indeed, that software is entirely resident in any one place rather than distributed.

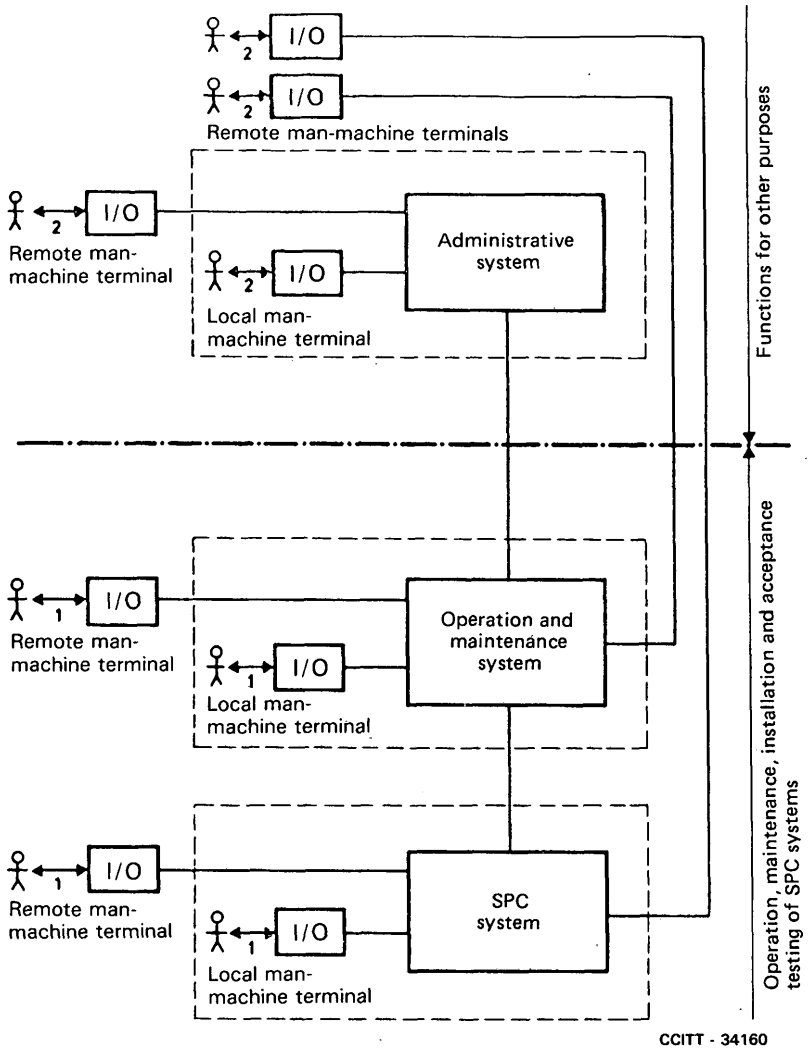
Although telephone signalling and switching has been considered the primary application area for the MML, these Recommendations accommodate the extension of the MML into other areas such as data switching, ISDN operations and maintenance, and software development environments.

In the Recommendations of this Part, the term *man* is used in the sense of *user*, and the terms *machine* and *system* are used interchangeably.

2 Man-machine communication model

Man-machine communication, the means of exchanging information between users and systems, can be represented by a layered model in which each layer defines features that support such communication. In their entirety, these features offer users an appropriate man-machine interface. The model is shown in Figure 2/Z.301 where higher layers are based upon features offered by the lower layers. The man-machine interface, for any given system, represented by the highest layer of the model, is based on the repertoire of inputs, outputs, special actions and man-machine interaction mechanisms, including dialogue procedures made available by the layers below.

These features are, in turn, supported by the lower layers in which the semantics associated with each MML function (actions, objects, information entities and their interrelationships) and the MML syntax are defined. The lowest layer of the model is identified in the set of system functions to be controlled and in the capabilities available in the man-machine terminals connected to the system.



1 Man-machine terminals where CCITT MML is recommended to be used.
2 Man-machine terminals where the use of CCITT MML is not considered.
I/O Input/output.

FIGURE 1/Z.301

Field of application of CCITT MML

Man-machine interface	
Inputs, outputs, special actions	Man-machine interaction mechanisms, including dialogue procedures
MML function semantics	MML syntax
System functions	Terminal capabilities

FIGURE 2/Z.301

Man-machine communication model

3 Organization of the MML Recommendations

The Recommendations on man-machine language are grouped in five sections:

1 General principles

2 Basic syntax and dialogue procedures

3 Extended MML for visual display terminals

4 Specification of the man-machine interface

5 Glossary of terms.

Section 1 gives an introduction to man-machine communication by the CCITT MML and contains information of a general nature. *Section 2* deals with syntax and dialogue procedures for terminals where no advantage is taken or can be taken of enhanced input and output facilities which are usually available on visual display terminals (VDTs). *Section 3* describes capabilities of VDTs and kinds of dialogue elements suitable for conveying the syntax of any application, including the syntax specified in Section 1, which can be applied to the operation and maintenance of SPC systems. As terminal technology advances and the theory of the man-machine interface evolves, greatly improved interfaces are possible. On the other hand, basic terminals will remain in use. Therefore this section provides a framework that accommodates interfaces possible on more sophisticated terminals and at the same time ensures that syntactic details presented at both sophisticated and basic terminals in a given application are consistent. *Section 4* identifies operation, maintenance, installation and acceptance testing functions to be controlled by the MML. A methodology is defined by which the semantics relating to MML functions may be generated and by which the inputs, outputs and special actions may be specified; specific Recommendations on Subscriber Administration, Routing Administration, Traffic Measurements Administration, and Network Management Administration are included. *Section 5* contains a summary of the terms used in Sections 1 to 4 together with short definitions to aid the reader seeking an explanation of a term.

4 Organization of Section 1

Section 1 consists of two Recommendations:

Z.301 Introduction to the CCITT man-machine language

Z.302 The meta-language for describing MML syntax and dialogue procedures.

Recommendation Z.302 enables the reader to interpret the diagrams used to specify MML syntax and dialogue procedures in Sections 2 and 3.

5 Basis of MML

The MML contains features which are sufficient to ensure that all relevant functions for the operation, maintenance, installation and acceptance testing of SPC systems can be performed.

The basic attributes of the language are summarized in the following:

a) The MML provides a consistent interface which is easy to learn and easy to use by novices as well as by experts, making possible the input of commands and the interpretation of outputs in a way convenient to all users.

b) The MML is flexible, allowing system design to be optimized according to the tasks to be performed. It offers a variety of input/output features including direct input, menus and forms.

c) The MML is adaptable to different kinds of personnel and to different national languages and organizational requirements.

d) The MML is structured to allow graceful incorporation of new technology.

The MML should be sufficiently flexible to meet Administrations' requirements for the organization of their operation and maintenance staff and for the security of their SPC systems; it should not restrict their selection of terminal types. The MML covers the man-machine interface including those functions that are initiated by the system and those that are initiated by the user. It should be implemented in such a way that errors in commands or control actions will not cause the system to stop, unduly alter the system configuration or take up undue resources.

6 Input/output

As indicated in Figure 1/Z.301, the interface being recommended is that between the user and an I/O device or devices. These devices must at least be capable of handling the code of the characters of the CCITT International Alphabet No. 5 both for input and for visual textual output to the user. Input will normally be from a keyboard device, but for bulk input of data and/or commands, some temporary storage medium such as paper tape, cassette, disc, etc., could be used. For output, a variety of device types is possible, including paper tape punches, teletypewriters, line printers, visual display terminals, etc.

7 Extensibility and sub-setting

The MML has an open-ended structure such that the addition of any new function or requirement will have no influence on the existing ones.

The language structure is such that sub-sets can be created. Sub-setting may be for various purposes, e.g., staff sub-sets, in which selection is done to meet the needs of certain sections of staff; application sub-sets, in which selection is made for convenience of application, etc.

Recommendation Z.302

THE META-LANGUAGE FOR DESCRIBING MML SYNTAX AND DIALOGUE PROCEDURES

1 Introduction

Syntax diagrams are a method of defining language syntax[1]. A syntax diagram consists of terminal and non-terminal symbol boxes connected by flowlines. An annotation symbol is used to insert comments. The syntax of a language can be defined by a series of syntax diagrams, each diagram defining a particular non-terminal symbol. In the MML Recommendations, syntax diagrams are used to assist in specifying the syntax of the MML input, MML output and the user-system dialogue procedures. A path through a syntax diagram defines an MML input, an MML output or a man-machine dialogue structure.

The sequence of symbols in a path through syntax diagrams does not always imply a corresponding order in time or in place. The order in time is only significant in dialogue procedures for changes in the direction of the information flow, i.e. from input to output or from output to input. For output on printers it represents an order in place (from left to right and from top to bottom). However, for output on VDTs, the order in place only applies to positions within a screen window (see Recommendation Z.322).

The following describes the use of syntax diagrams and states a set of rules for their use.

2 Terminology

2.1 Terminal symbols are those characters or strings of characters which actually appear in the input or output. To avoid possible misunderstanding, format effectors are represented by a crossed mnemonic of the intended format effector.

2.2 A non-terminal symbol does not immediately appear in MML input or MML output; it represents, within a syntax diagram, another syntax diagram by name. Hence it is an abbreviated symbol for a more complex construct (consisting of a set of terminal and/or non-terminal symbols) used in several places.

2.3 Annotation symbols (see § 3.7) are used to insert references to descriptive or explanatory notes. For example, they may be used to indicate mutually exclusive paths through a diagram.

---

[1] The syntax diagrams used in MML are based on those used to describe the programming language PASCAL [1].

3 Rules

3.1 Every symbol box (terminal or non-terminal) and consequently each diagram must have one, and one only, entry and one, and one only, exit flowline.

3.2 Each diagram must fit on a single page. Thus there is no off-page connector symbol.

3.3 Flowlines are always unidirectional. The preferred direction for flowlines which select alternatives is down. The preferred direction for flowlines which connect symbols is left-to-right. The preferred direction for flowlines which indicate repetitions (loops) is counterclockwise.

3.4 An arrowhead is required wherever any two flowlines come together, and wherever a flowline enters a symbol box. Additional arrowheads may be inserted wherever it is felt that this will improve the clarity of the diagram.

3.5 Terminal symbols are surrounded by boxes with rounded corners. The width of the box is proportional to the number of characters contained in the box. For short terminal symbols, the box may become a circle. Symbols representing system input are surrounded by a single solid line and those representing system output by a double solid line:

- for terminal input symbols see Figure 1a)/Z.302 and Figure 1b)/Z.302,
- for terminal output symbols see Figure 1c)/Z.302 and Figure 1d)/Z.302.

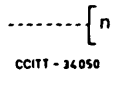
3.6 Non-terminal symbols are surrounded by rectangular boxes. The name of the non-terminal symbol must be written in lower case characters. Every non-terminal symbol must have an associated syntax diagram except where the symbol is annotated "Not further expanded in diagram form". The non-terminal symbol used to name a particular syntax diagram must appear at the upper left corner of the diagram. Symbols representing system input are surrounded by a single solid line, those representing system output by a double solid line and symbols representing a combination of input and output by an outer solid and an inner dashed line:

a) for the non-terminal input symbol see Figure 1e)/Z.302,

b) for the non-terminal output symbol see Figure 1f)/Z.302,

c) for the non-terminal input/output symbol used in dialogue procedures see Figure 1g)/Z.302.

3.7 An annotation is denoted by the following symbol:



where n is a number referring to a descriptive or explanatory note. The text of the note must be located at the foot of the diagram.

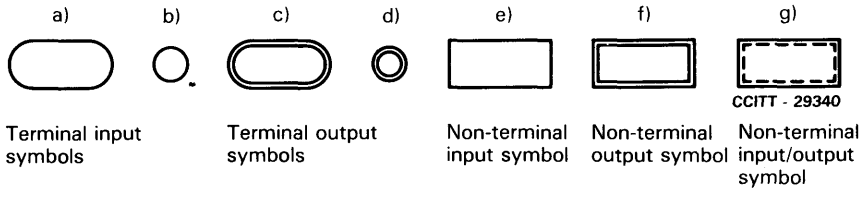


FIGURE 1/Z.302

Terminal and non-terminal symbols to be used for the CCITT man-machine language

Reference

[1] JENSEN (K.), WIRTH (N.): PASCAL, User Manual and Report, *Springer Verlag*, New York, 1975.

PAGE INTENTIONALLY LEFT BLANK

PAGE LAISSEE EN BLANC INTENTIONNELLEMENT

SECTION 2

BASIC SYNTAX AND DIALOGUE PROCEDURES

Recommendation Z.311

INTRODUCTION TO SYNTAX AND DIALOGUE PROCEDURES

1 Scope of the Section

Section 2 deals with syntax and dialogue procedures for terminals where no advantage is taken or can be taken of enhanced input and output facilities which are usually available on VDTs. This basic MML is therefore compatible with the use of VDTs used in the manner of teletypewriters, hard copy printers, etc., at the man-machine interface.

2 Organization of Section 2

Section 2 consists of the following Recommendations:

Z.311 Introduction to syntax and dialogue procedures
Z.312 Basic format layout
Z.313 (spare)
Z.314 The character set and basic elements
Z.315 Input (command) language syntax specification
Z.316 Output language syntax specification
Z.317 Man-machine dialogue procedures.

Recommendation Z.317 describes the operational procedures for a dialogue between user and system. For aspects of input syntax, reference is made to *Recommendation Z.315* and for aspects of output syntax, reference is made to *Recommendation Z.316.* Recommendation Z.316 also deals with output outside dialogue. The specification of basic syntax elements for input and output, together with the characters to be used, is contained in *Recommendation Z.314.* Formats to be used on teletypewriters and hard copy printers are described in *Recommendation Z.312.*

Recommendation Z.312

BASIC FORMAT LAYOUT

1 General

To facilitate filing and retrieval of recorded information in MML, it is recommended that this information should be recorded on pages with an identification header at the top of each page. The top and bottom lines of a page should not be used but should be left blank.

It is further recommended that the layout for printing information in MML should be based on a maximum of 72 characters per line and 66 lines per page, as this format can be accommodated on both the A4 and the 11-inch standard size paper and can be printed by standard teletypewriters.

Where a number of characters per line in excess of 72 is required, a second format is recommended. This accommodates 120 characters per line and would be used, for example, on line printers.

In order to save paper or where paging to facilitate filing of output is not required, paging may be suppressed by suppressing the generation of all superfluous line feeds.

To distinguish between the formats recommended, they are further indicated as format F1 for the paper sizes A4 and A5L and format F2 for the paper size A4L. In the recommended formats specified below, International Standard ISO/2784 [1] has been taken into account.

2 Recommended formats for presenting information in MML

2.1 *Format F1*

According to this format, which is based on the A4 and the 11-inch standard size paper, the maximum number of characters per line is 72. The number of lines per page may be 66, using the full 11-inch and A4 paper sizes or 33, using half the paper size (5.5 inch or A5L).

Information presented in this format can also be displayed on most of the VDTs available on the market. However, the number of lines which can be displayed at the same time on these devices is, in general, not more than 20 to 25 lines.

2.2 *Format F2*

This format allows a maximum of 120 characters printed on a line and has 66 lines per page. It can be accommodated on paper having a width equal to the A4L standard.

Reference

[1] International Organization for Standardization: *Continuous forms used for information processing. Sizes and Sprocket feed holes*, ISO 2784-1974.

Recommendation Z.314

THE CHARACTER SET AND BASIC ELEMENTS

1 General

The character set and the basic elements used in the syntax are essential components of MML inputs, MML outputs and the man-machine dialogue procedures.

2 The character set

The character set to be used for the CCITT MML is a sub-set of the CCITT International Alphabet No. 5 which has been established jointly by the CCITT and the International Organization for Standardization.

To allow for possible implementation of the CCITT MML using national languages, the sub-set is taken from the basic code table given in Recommendation T.50 [1] The code positions reserved in this table for national use are not contained in the basic character set of the CCITT MML, but may be used in these national implementations.

According to Recommendation T.50 [1] transmission control characters and information separators are intended to control or to facilitate transmission of information over telecommunication networks. Hence these control characters are not used in the MML. This will avoid interference with data transmission procedures when information in the MML is transmitted via a data transmission network.

It is furthermore recommended when information is printed or displayed that devices are used which print or display different graphic symbols for the digit zero and the capital letter O.

The characters selected for use in the CCITT MML are given in Table 1/Z.314.

TABLE 1/Z.314

Character set to be used for the CCITT man-machine language

				b_7	0	0	0	0	1	1	1	1
				b_6	0	0	1	1	0	0	1	1
				b_5	0	1	0	1	0	1	0	1
b_4	b_3	b_2	b_1	Pos.	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL		SP	0	ⓐ	P	ⓐ	p
0	0	0	1	1		DC_1	!	1	A	Q	a	q
0	0	1	0	2		DC_2	"	2	B	R	b	r
0	0	1	1	3		DC_3	#	3	C	S	c	s
0	1	0	0	4		DC_4	¤	4	D	T	d	t
0	1	0	1	5			%	5	E	U	e	u
0	1	1	0	6			&	6	F	V	f	v
0	1	1	1	7	BEL		'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT (FE1)	EM	)	9	I	Y	i	y
1	0	1	0	10	LF (FE2)	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT (FE3)	ESC	+	;	K	ⓐ	k	ⓐ
1	1	0	0	12	FF (FE4)		,	<	L	ⓐ	l	ⓐ
1	1	0	1	13	CR (FE5)		-	=	M	ⓐ	m	ⓐ
1	1	1	0	14	SO		.	>	N	ⓐ	n	ⓐ
1	1	1	1	15	SI		/	?	O	_	o	DEL

CCITT - 26622

ⓐ These positions are reserved for national use.

General remarks – The characters proper to the open positions are considered as outside the MML. They are implementation dependent and, together with the characters named in the table but not included in the MML, may be used in accordance with the rules given in Recommendation T.50 [1]. The position of a character in the table can be indicated by its column and row number, e.g., Pos. 3/1 gives the position of the digit 1 in the table. The table gives also the binary codes allocated to the table positions according to Recommendation T.50 [1]. The bits are identified by b_7, b_6, ... b_1, where b_7 is the highest order, or most significant bit, and b_1 is the lowest order, or least significant bit.

3 Summary of use of characters

The use of characters in the character set, except for letters, digits, and characters used solely as graphic characters and format effectors, is described in Table 2/Z.314. CCITT International Alphabet No. 5 code is indicated by position number (see Table 1/Z.314).

3.1 *Letter*

A letter is one of the characters listed in Table 1/Z.314, columns 4, 5, 6 and 7. However, positions 5/15 and 7/15 are excluded. The characters reserved for national use may be used as letters or as graphic characters.

3.2 *Digit*

A digit is one of the characters listed in Table 1/Z.314, column 3, positions 0 to 9.

3.3 *Graphic characters*

Graphic characters are a collection of characters one or more of which may be used to improve readability. Graphic characters which have other syntactic uses are listed in Table 2/Z.314. The $ (position 2/4 in Table 1/Z.314) is the only character used solely as a graphic character.

3.4 *Format effector*

The format effectors used in MML are the characters FE1 to FE5 and SP (space) as defined in Table 1/Z.314. The character BACK SPACE (FE0 in Recommendation T.50 [1]) is not regarded as a format effector in the MML.

4 Basic elements used in the syntax

Syntax diagrams of the basic elements used in the syntax are given in § 5 in sub-paragraphs with numbers corresponding to those in § 4.

4.1 *Identifier*

An identifier is a string of one or more characters which begins with a letter and, if applicable, subsequently contains only digits and/or letters e.g., U, UPDATE, UPD8.

4.2 *Symbolic name*

A symbolic name is a string of one or more characters used for the purpose of representing an entity which cannot be adequately represented by numerals or identifiers. The string contains at least one letter and/or at least one of the graphic characters + (plus sign), # (number sign), % (percent sign) plus any number of digits, including none. The characters may appear in any order. For example a time duration of 6 hours may be represented by the symbolic name 06H, a 10 percent threshold value by 10%, a signalling system such as CCITT No. 6 by SS#6.

4.3 *Decimal numeral*

A decimal numeral is a character combination, consisting of a digit or digits and an optional . (full stop), preceded by the special character combination D' (D apostrophe). If the numeric default base for an information unit (see Recommendation Z.315) is decimal, then the D' is optional.

4.4 *Nondecimal numerals*

A nondecimal numeral is a character combination preceded by a special character combination indicating the type of numeral.

4.4.1 H' (H apostrophe) is used to indicate a hexadecimal numeral, the following characters thus being any of: digits 0 to 9 or letters A, B, C, D, E, F.

4.4.2 O' (letter O apostrophe) is used to indicate an octal numeral, the following characters thus being any of: digits 0, 1, 2, 3, 4, 5, 6, 7.

4.4.3 B' (B apostrophe) is used to indicate a binary numeral, the following characters thus being digit(s) 0 and/or digit(s) 1.

4.4.4 K' (K apostrophe) is used to indicate a keyed numeral, the following characters thus being any of: digits 0-9, * (asterisk), # (number sign), or letters A, B, C, D.

4.4.5 When the default base for an information unit (see Recommendation Z.315) is one of the nondecimal numerals e.g., hexadecimal, the corresponding character combination, i.e. H' in this example, is optional.

TABLE 2/Z.314

Summary of use of characters

CCITT International Alphabet No. 5 (Recommendation T.50) [1]			Man-machine language use
Character or character string	Position number	Name	
CAN	1/8	Cancel	Used as a deletion character.
!	2/1	exclamation mark	An indicator used in dialogue procedures (continuation character in input language).
"	2/2	quotation mark	A text string delimiter and a graphic character.
#	2/3	number sign	A character which may be used in symbolic names and keyed numerals and as a graphic character.
%	2/5	percent sign	A character which may be used in symbolic names and as a graphic character.
&	2/6	ampersand	A separator for information grouping and a graphic character.
'	2/7	apostrophe	A separator used when indication of type of numeral is required. The character is placed between a letter indicating the type of numeral and the numeral itself. Also used as a graphic character.
(	2/8	left parenthesis	Used for delimiting arithmetical expressions, and conditions in a selection argument. Also a graphic character.
)	2/9	right parenthesis	Used for delimiting arithmetical expressions, and conditions in a selection argument. Also a graphic character.
*	2/10	asterisk	Used for keyed numerals, as an arithmetic operator and as a graphic character.
+	2/11	plus sign	A character which may be used in symbolic names, as an arithmetic operator and as a graphical character.
+ +	2/11 2/11	plus sign, plus sign	A separator used for separating the increment from a group of consecutive parameter values.
,	2/12	comma	A separator used to separate parameters (if more than one) within a block of parameters.
–	2/13	hyphen	A separator used to separate information units or to separate identifiers and/or index numbers within compound parameter names. Also used as an arithmetic operator and as a graphic character.
.	2/14	full stop	A separator used for subdividing a number into an integer part and a fraction part and as a graphic character.
/	2/15	solidus	Used as an arithmetic operator and as graphic character.
:	3/10	colon	A separator used to separate blocks of parameters from each other and from the command code, an indicator used in the parameter block request indication and a separator used in output.
;	3/11	semicolon	An indicator used to terminate a command (execution character).
<	3/12	less than sign	An indicator used as a ready indicator for the system to output that it is ready to receive information, and a relational operator used in a selection argument.
=	3/13	equal sign	A separator used to separate the parameter name and the parameter value of a parameter. Also a relational operator used in a selection argument.

TABLE 2/Z.314 *(cont.)*

CCITT International Alphabet No. 5 (Recommendation T.50) [1]			Man-machine language use
Character or character string	Position number	Name	
>	3/14	greater than sign	A separator to terminate the destination identifier and a relational operator used in a selection argument.
< =	3/12 3/13	less than or equal sign	A relational operator used in a selection argument.
< >	3/12 3/14	less than or greater than sign	A relational operator used in a selection argument.
> =	3/14 3/13	greater than or equal sign	A relational operator used in a selection argument.
?	3/15	question mark	An indicator used for prompting or help.
&&	2/6 2/6	ampersand, ampersand	Separator used for information grouping.
&–	2/6 2/13	ampersand, hyphen	Separator used for information grouping.
&&–	2/6 2/6 2/13	ampersand, ampersand, hyphen	Separator used for information grouping.
/*	2/15 2/10	solidus, asterisk	Used to open a comment.
*/	2/10 2/15	asterisk, solidus	Used to close a comment.

4.5 *Text string*

A text string allows input of a literal text, including any delimiters which would have syntactical meanings when input outside a text string. It consists of a string of zero or more characters enclosed by a " (quotation mark) at the beginning and end. The string may contain any of the characters belonging to the character set defined in § 2 except correction characters (see Recommendation Z.315). If " (quotation mark) is required within a string, it is represented by "" (double quotation marks). Text strings in output need not be delimited by quotation marks. Text strings for use in extended MML (Recommendations Z.321-Z.323) need not be delimited by quotation marks.

4.6 *Arithmetical expression*

An arithmetical expression is a combination of certain basic elements and arithmetic operators delimited by parentheses.

4.7 *Ancillary facilities*

Additional facilities have been provided when using MML commands as follows.

4.7.1 *Comment facility*

A comment is defined as a character string enclosed between the separators /* (solidus asterisk) and */ (asterisk solidus), where the character string may contain any characters except the sequence */ (asterisk solidus) and correction characters (see Recommendation Z.315). The character string, including the delimiters, has neither

MML syntactical nor semantical significance. However, if it occurs in a text string, it is regarded as being part of the text string. A comment may be inserted only before and/or after a separator, indicator, arithmetic delimiter [((left parenthesis),) (right parenthesis)], arithmetic operator [+ (plus sign), - (hyphen), / (solidus), * (asterisk)], identifier or information unit [excluding the ' (apostrophe) between the type of numeral and the numeral itself and the . (full stop) between the integer and fractional part of a number].

4.7.2 *Escape syntax*

In some systems it is not possible to use characters with syntactical meaning [e.g., ; (semi colon), - (hyphen)] or correction characters as data. In such systems an escape indication may be used in order to introduce the following character as data.

A specific escape indication is not proposed due to the diverse nature of terminals.

No syntax diagram is given.

4.7.3 *Format effector*

A format effector (see § 3.4) is used to format input and output in a suitable manner. Format effectors have no significance in a command and may appear anywhere in input.

No syntax diagram is given.

4.8 *Separator*

A separator is a character or a string of characters used to separate items of information in the input or output and it may, in addition, have structural, semantic or other significance.

No syntax diagram is given.

4.9 *Indicator*

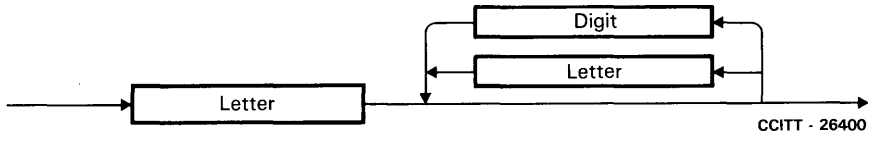
An indicator is a character used to indicate a state or make a request.

No syntax diagram is given.

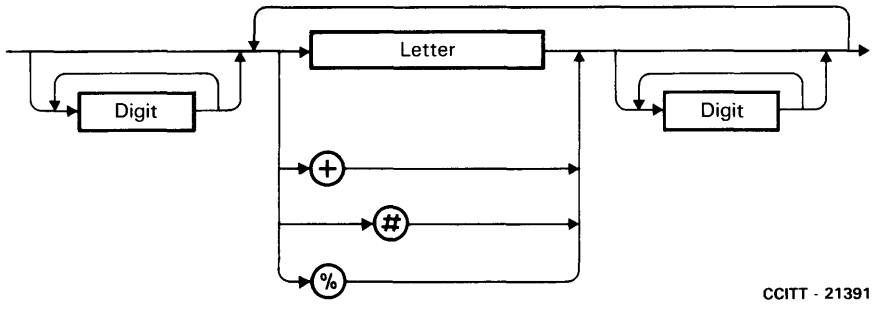
5 Definition of the basic elements used in the syntax in diagrams

All these elements may be used in both input and output but for simplicity only the input elements are shown in the diagrams. The output elements are identical to the input elements.

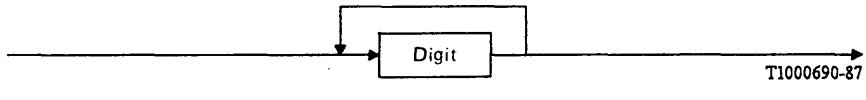
5.1 *Identifier*



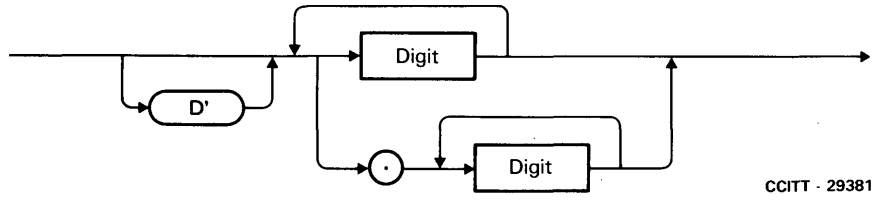
5.2 *Symbolic name*



5.3 *Index number*

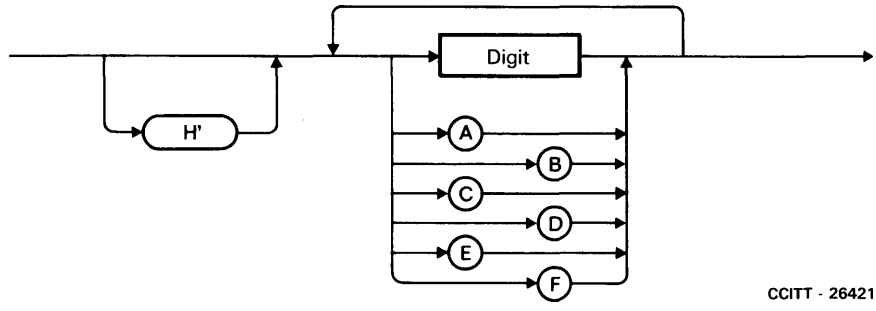


5.4 *Decimal numeral*

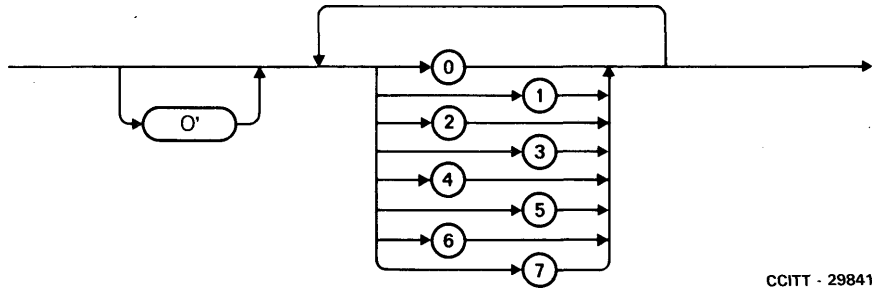


5.5 *Nondecimal numerals*

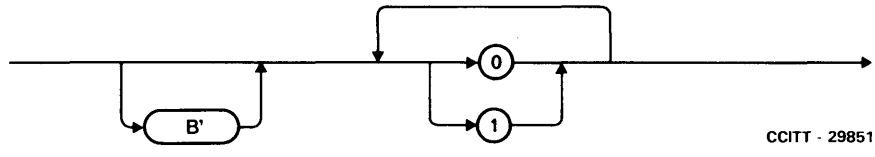
5.5.1 *Hexadecimal numeral*



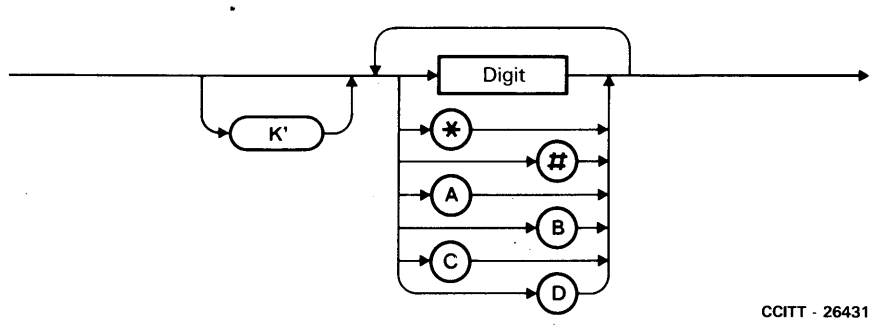
5.5.2 *Octal numeral*



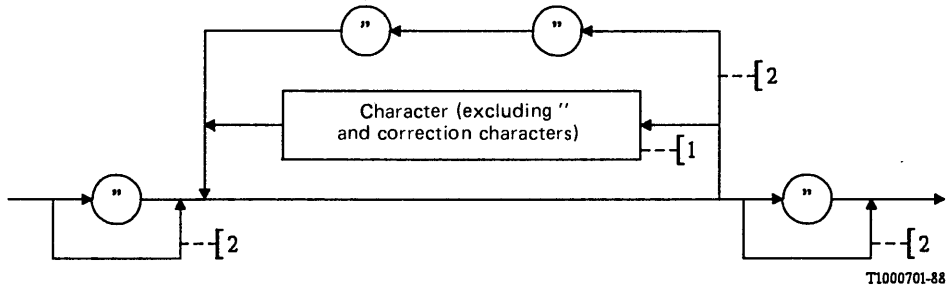
5.5.3 *Binary numeral*



5.5.4 *Keyed numeral*



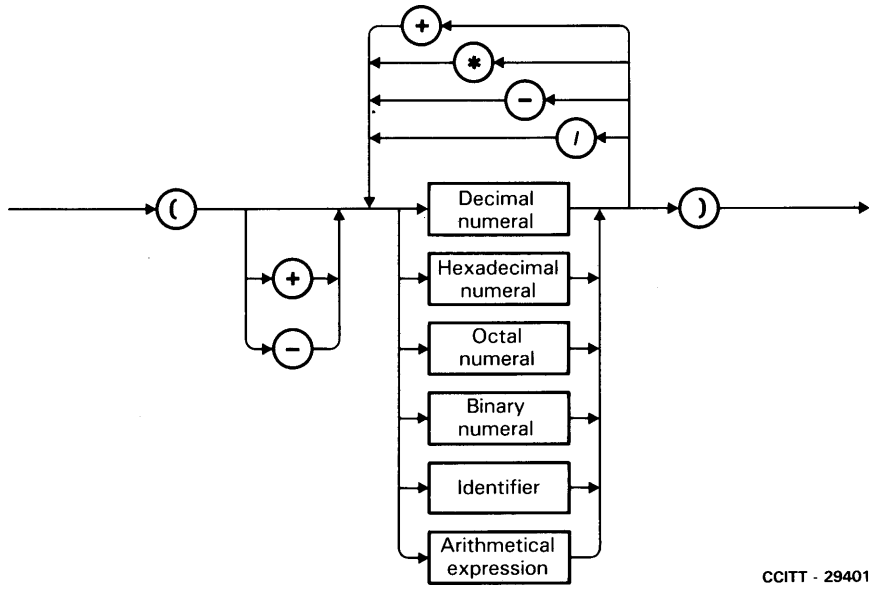
5.6 *Text string*



1) Not further expanded in diagram form.

2) When using this diagram for output in extended MML (Recommendations Z.321-Z.323), text strings need not be delimited with quotation marks.

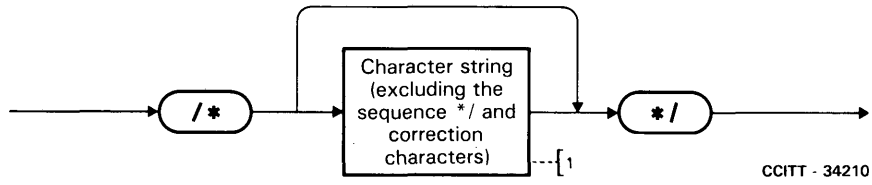
5.7 *Arithmetical expression*



Note – The deepest level of the arithmetical expression has to satisfy the diagram in a form where the box "arithmetical expression" is omitted.

5.8 *Ancillary facilities*

5.8.1 *Comment*



1) Not further expanded in diagram form.

Reference

[1] CCITT Recommendation *International Alphabet No. 5*, Rec. T.50.

Recommendation Z.315

INPUT (COMMAND) LANGUAGE SYNTAX SPECIFICATION

1 General

The following text describes the elements of the input language. Syntax diagrams of the input language are given in § 4 in sub-paragraphs with numbers corresponding to those in § 2. Where input elements are used in output, reference to these elements is made in the output language description Recommendation Z.316. Procedural aspects are taken into account in Recommendation Z.317. It should be noted that certain areas of the syntax allow options to be taken which could result in a syntax clash. The taking of such options must be chosen to suit the particular system involved.

2 Command structure

2.1 *Command*

A command begins with the command code, which defines the function to be performed by the system. If further information is required a command code can be followed by a parameter part from which it is separated by a : (colon). The parameter part consists of one or more blocks of parameters (see §§ 2.3 and 2.9.1). A command is always completed by an execution character (see Recommendation Z.317).

2.2 *Command code*

The command code is composed of up to three identifiers separated by a - (hyphen) (e.g., functional area - object type - action). Where command codes are in the form of single mnemonic abbreviations, it is recommended that they consist of the same number of characters.

2.3 *Block of parameters*

A block of parameters contains information necessary to execute the function specified in the command code. The information in a block of parameters is expressed in the form of a number of parameters specific to the command. If more than one parameter is included, they shall be separated by a , (comma). All parameters in any one block shall be of the same kind i.e. either position defined parameters or parameter name defined parameters.

2.4 *Parameters*

A parameter identifies and contains a piece of information and may be either position defined or parameter name defined. Non-relevant parameters may be omitted in accordance with §§ 2.4.1 and 2.4.2.

2.4.1 *Position defined parameter*

A position defined parameter consists of a parameter value which may be preceded by a parameter name from which it is separated by an = (equal sign). Parameters must be given in a predetermined order within the parameter block. Where a parameter value is not to be given, the parameter is omitted leaving the appropriate separator or the appropriate indicator used to terminate a command. This indicates the parameter's position in the block of parameters. Parameter omission can imply that the default value is meant. The default value can also be indicated by giving a parameter value assigned for this purpose.

2.4.2 *Parameter name defined parameter*

A parameter name defined parameter consists of a parameter name followed by a parameter value from which it is separated by an = (equal sign). These parameters may be given in an arbitrary order within the parameter block. Where a parameter value is not to be given, the parameter name and separator = (equal sign) and the separator , (comma) following the parameter are also omitted. This omission can imply that the default value is meant. The default value can also be indicated by giving a parameter value assigned for this purpose. Where a parameter value implies the parameter name the latter and the separator = (equal sign) can be omitted.

2.5 *Parameter name*

A parameter name unambiguously indicates the kind and structure of the subsequent parameter value and thereby defines the parameter value and how it shall be interpreted. It is an identifier. It is either a simple parameter name or a compound parameter name. The simple parameter name indicates a single parameter value and a compound name indicates a parameter value in a list or table of similar parameter types.

2.5.1 *Simple parameter name*

A simple parameter name consists of one identifier.

2.5.2 *Compound parameter name*

A compound parameter name consists of one or more identifiers and/or index number all separated by a separator - (hyphen).

2.5.2.1 *Index number*

An index number is one or more digits.

2.6 *Parameter value*

A parameter value contains the information required to specify the appropriate object(s) or value(s) and consists of one or more information units. In the case where no information grouping (see § 2.9) is applied a parameter value reduces to a parameter argument. Refer to § 2.10 for data base query aspects.

2.7 *Parameter argument*

A parameter argument contains the information required to specify the appropriate object or value. It is the form of a parameter value when no information grouping is applied (see § 2.9). A parameter argument consists of a simple or a compound parameter argument.

2.7.1 *Simple parameter argument*

A simple parameter argument consists of one information unit.

2.7.2 *Compound parameter argument*

A compound parameter argument consists of two or more information units separated by a - (hyphen).

2.8 *Information unit*

An information unit constitutes the smallest unit of information in the language from a syntactical point of view. An information unit can be a numeral, an identifier, a symbolic name, a text string or an arithmetical expression. A numeral always has a default base (e.g., hexadecimal) which can be overwritten, if required, by introducing the desired base as specified in Recommendation Z.314. However, the default base for a keyed numeral cannot be overwritten by another base.

2.9 *Information grouping*

Information grouping is used to improve the speed and ease of input activities. It is performed by grouping sets of information of the same type within the same command.

2.9.1 *Grouping of blocks of parameters*

If several blocks of parameters are to be included in one command they shall be separated by a : (colon).

2.9.2 *Grouping of parameter arguments*

Input of more than one parameter argument within one parameter of a command can be achieved by grouping parameter arguments.

2.9.2.1 *Grouping of simple parameter arguments*

It is possible to indicate several simple parameter arguments within the same parameter value separated by an & (ampersand). *Example 1:* 5&9 means the simple parameter arguments 5 and 9.

In the case of a sequence of consecutive (implicit increment value = 1) simple parameter arguments, it is possible to indicate the arguments by writing the lower and upper simple parameter arguments separated by an && (ampersand ampersand)[1]. *Example 2:* 5&&9 means the simple parameter arguments 5, 6, 7, 8 and 9.

An explicit increment value can be specified following the upper parameter argument separated by + + (plus plus). *Example 3:* 5&&9+ +2 means the simple parameter arguments 5, 7 and 9.

Other combinations of the above possibilities may also be used when required. *Example 4:* 5&&7&9 means the simple parameter arguments 5, 6, 7 and 9. *Example 5:* 5&&9+ +2&10 means the simple parameter arguments 5, 7, 9 and 10.

2.9.2.2 *Grouping of compound parameter arguments*

It is possible to indicate several compound parameter arguments within the same parameter value separated by an & (ampersand). *Example 1:* 5-1&6-3 means the two compound parameter arguments 5-1 and 6-3.

If a group of compound parameter arguments differs only in the last information unit, the first compound parameter argument is completely specified, whereas all subsequent compound parameter arguments are represented only by their last information units, separated by an &- (ampersand hyphen). *Example 2:* 7-1&-3 means the two compound parameter arguments 7-1 and 7-3.

If a group of compound parameter arguments differs only in the last information unit and constitutes a consecutive sequence (implicit increment value = 1), it is possible to indicate the arguments by writing the lower and upper information units separated by an &&- (ampersand ampersand hyphen)[1]. *Example 3:* 7-1&&-3 means the three compound parameter arguments 7-1, 7-2 and 7-3. *Example 4:* 7-1&-3&&-5 means the four compound parameter arguments 7-1, 7-3, 7-4 and 7-5.

---

[1] The interpretation of the separators && (ampersand ampersand) and &&- (ampersand ampersand hyphen) is not exclusive. Other interpretations exist. One alternative would imply that no specific increment is inherent in the syntax. That is, the relationship of the values between the upper and lower values in the sequence is a semantic relationship dependent upon the function for which the sequence is being specified.

An explicit increment value can be specified following the upper information unit separated by + + (plus plus).

Any combination of the above possibilities may also be applied when required. *Example 5:* 5-1&&-3&8-2&-5&-6 means the six compound parameter arguments 5-1, 5-2, 5-3, 8-2, 8-5 and 8-6. *Example 6:* 5-1&&-7 + +2&8-1&-3 means the six compound parameter arguments 5-1, 5-3, 5-5, 5-7, 8-1 and 8-3.

2.10 *Data base queries*

Data base queries are expressed in terms of projection and selection information. Projection information can be represented by a parameter. Its name identifies the projection function. Its group of parameter argument(s) identifies the appropriate field(s) of the data records to be displayed. Selection information can be represented by a parameter where the name identifies the selection function and the value identifies a (group of) selection argument(s). A selection argument comprises one or more conditions that should all be satisfied. A condition is specified by an identifier and a (group of) parameter argument(s) separated by a relational operator. The identifier specifies the name of the field and of the record to be selected. Omission of the selection information implies that the query is not conditional.

The names "projection" and "selection" are chosen for the example only. Other names such as "select" and "where" may apply.

Examples:

query-dbx: projection = field a,

selection = (field c = 0);

This command requests the records that satisfy the selection criterion field c = 0 of data set x; however, only field a of the selected records needs to be displayed.

query-dbx: projection = field a & field b,

selection = (field b > 5, field c = 1);

This command requests the records that satisfy both the selection criteria field b > 5 and field c = 1 of data set y. The resulting display need only show the fields a and b of the selected records.

query-dbz: projection = field a & field b & field d,

selection = (field d < = 7, field e = 0) & (field b = P);

This command requests from data set z the records that satisfy both the criteria field d< = 7 and field e = 0. It also requests the records that satisfy the criterion field b = P. The display of all selected records need only show fields a, b, and d.

Warning

The use of characters , (comma) and & (ampersand) in CCITT-MML corresponds to the operators AND and OR in predicate logic. A general assumption is made that predicate logic is not used by normal operating personnel. Confusion can also be avoided by realizing the functions of the various separation characters in CCITT-MML. The comma is used as a separator of parameters within a block, where all parameters together play a role in executing the command. The ampersand serves as a separator in information grouping, and is used to input one command "value1&value2", as an alternative to inputting two commands, one for "value1", and one for "value2".

Restriction

To avoid meaningless expressions, the parameter argument if used in combination with a non-symmetrical relational operator in syntax diagram 4.10.1.1 (condition) should be restricted to numerals. However, identifiers and symbolic names are allowed if they represent members of an ordered set.

3 Corrections and delete command

Corrections can be made by the deletion and resubmission of input.

Specific characters are not proposed because of the diverse nature of Input/Output terminal devices available.

3.1 *Delete last character*

The facility may be used to delete successive input characters back to the last system output (see § 3.2).

3.2 *Delete to last system output*

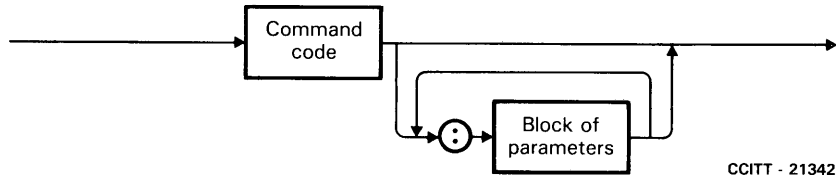
This facility deletes all input characters after the last system output, being either the ready indication or prompting output (see Recommendation Z.317).

3.3 *Delete command*

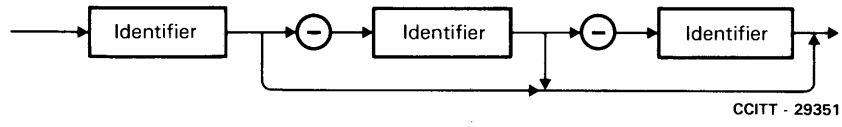
The delete command request is conveyed by the CAN character (cancel). The use of this character causes the system to respond with an acknowledgement that presents input after the last command executed is cancelled. The system should respond with a new ready indication to indicate that it is waiting for a new command code (see Recommendation Z.317).

4 Definition of the input (command) language structure in syntax diagrams

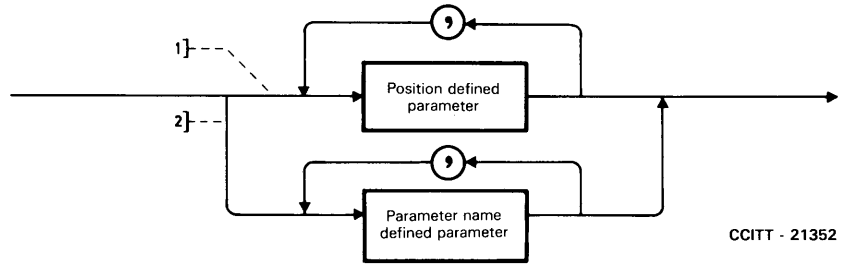
4.1 *Command*



4.2 *Command code*



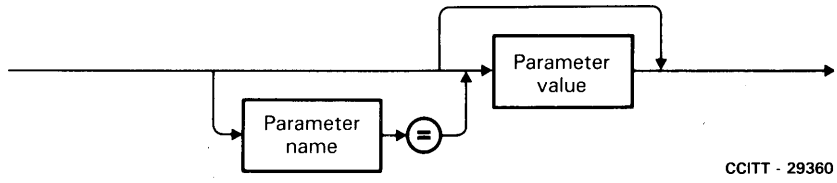
4.3 *Block of parameters*



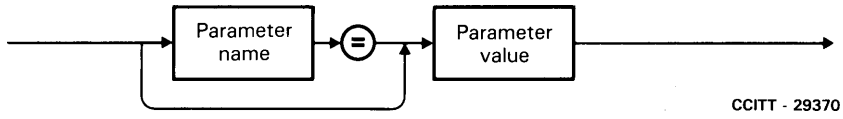
1) Upper main branch valid only for block of position defined parameters.
2) Lower main branch valid only for block of parameter name defined parameters.

4.4 *Parameters*

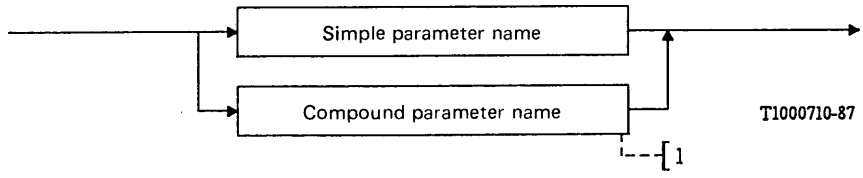
4.4.1 *Position defined parameter*



4.4.2 *Parameter name defined parameter*

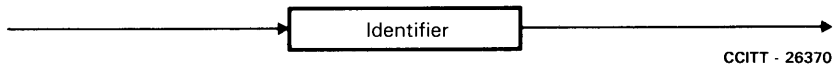


4.5 *Parameter name*

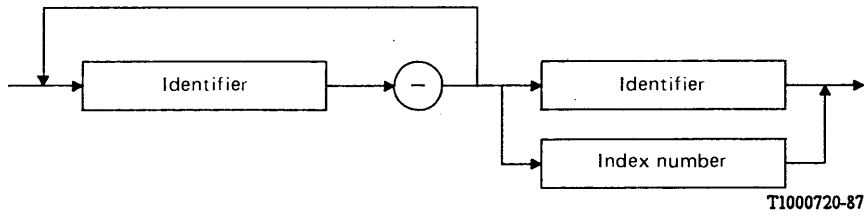


1) This is optional.

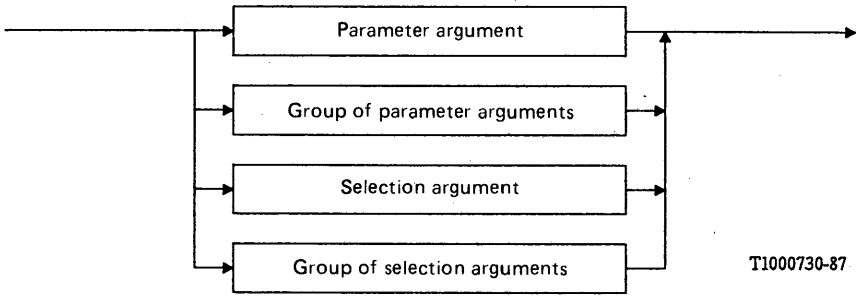
4.5.1 *Simple parameter name*



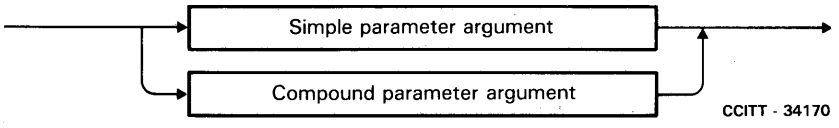
4.5.2 *Compound parameter name*



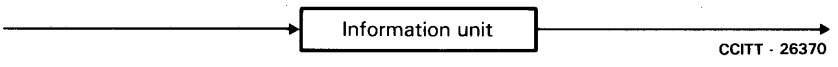
4.6 *Parameter value*



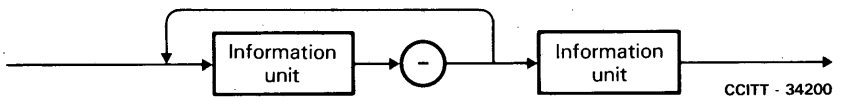
4.7 *Parameter argument*



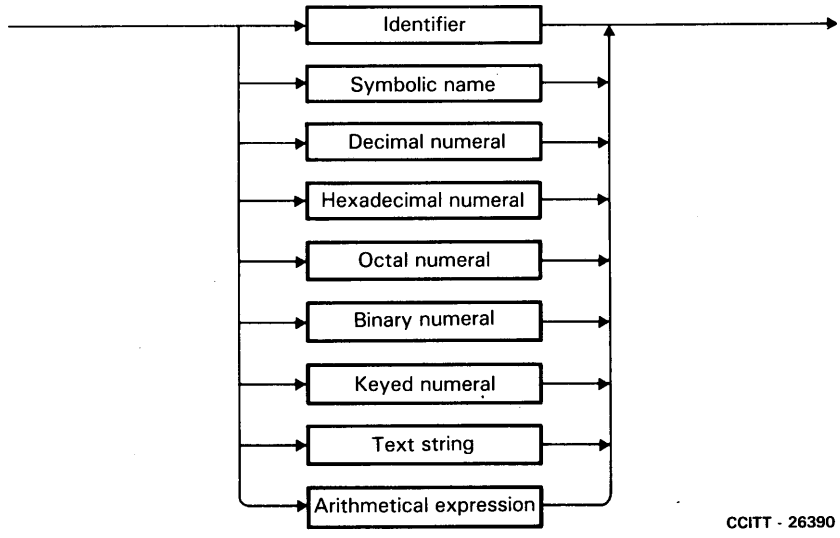
4.7.1 *Simple parameter argument*



4.7.2 *Compound parameter argument*



4.8 *Information unit*

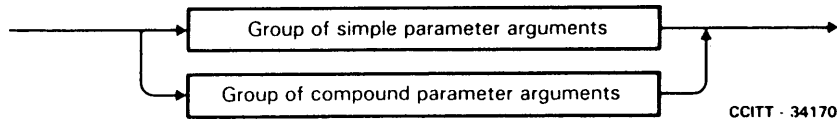


4.9 *Information grouping*

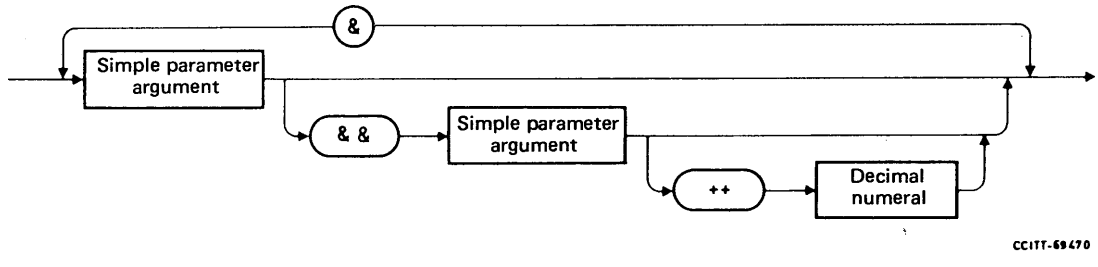
4.9.1 *Group of blocks of parameters*

See syntax diagram § 4.1.

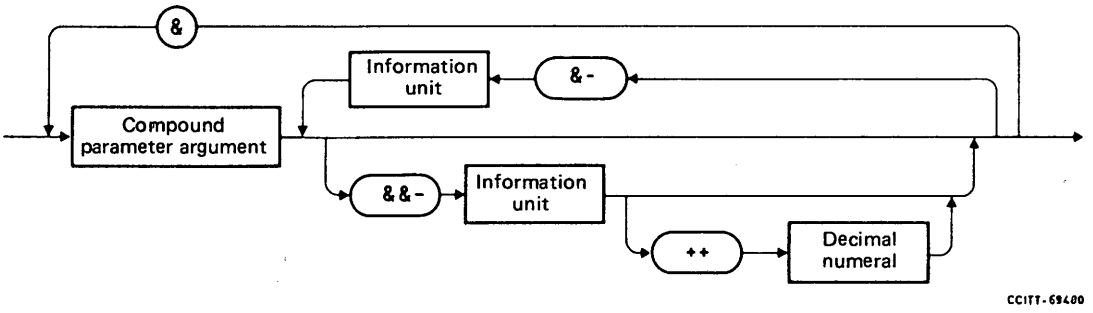
4.9.2 *Group of parameter arguments*



4.9.2.1 *Group of simple parameter arguments*

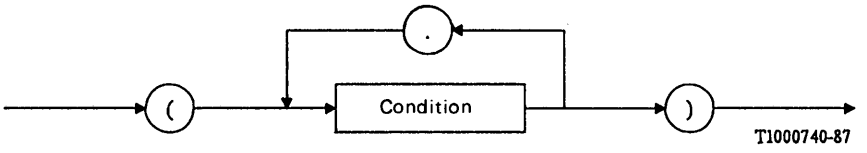


4.9.2.2 *Group of compound parameter arguments*

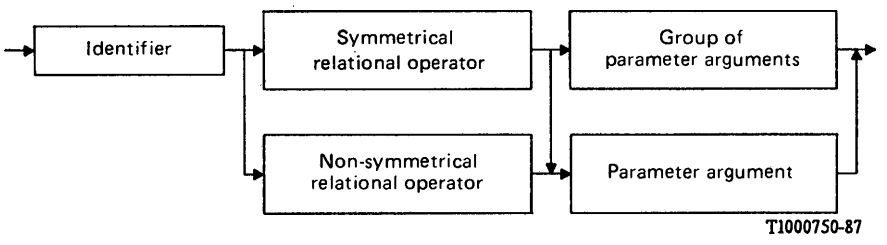


4.10 *Data base queries*

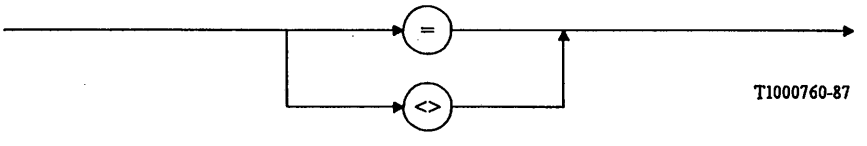
4.10.1 *Selection argument*



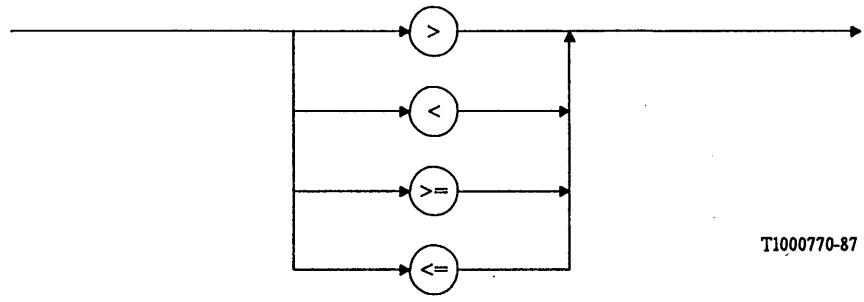
4.10.1.1 *Condition*



4.10.1.2 *Symmetrical relational operator*

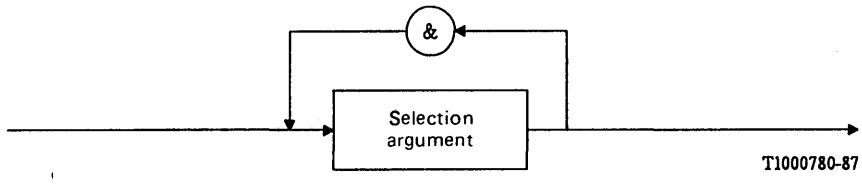


4.10.1.3 *Non-symmetrical relational operator*



T1000770-87

4.10.2 *Group of selection arguments*



T1000780-87

Recommendation Z.316

OUTPUT LANGUAGE SYNTAX SPECIFICATION

1 General

Syntax diagrams of the output language are given in § 3 in sub-paragraphs having numbers corresponding to those in § 2. Where input elements are used in output, a reference is made to the input language description Recommendation Z.315. Procedural aspects utilizing output other than output outside dialogue are taken into account in Recommendation Z.317.

2 Output structure

2.1 *Output outside dialogue*

The output described is output outside dialogue. This output is either a spontaneous output indicating a certain event, e.g., an alarm situation, or it is a delayed response to an interactive operating sequence (see Recommendation Z.317). An example of such a delayed response is a traffic measurement result.

2.2 *Header*

The header is given in output outside dialogue. It is also used in the dialogue procedure (see Recommendation Z.317). The main purpose of the header is to mark the output outside dialogue or the record of the dialogue for identification and information. The header can also be used for special purposes for an operation and maintenance centre. Recommended contents are information related to source identification, date and time. More information not related to the input or output function can be added to the header as additional header information.

The header is introduced by format effectors and/or graphic characters selected from a layout option.

2.2.1 *Layout option*

A layout option is a combination of format effectors and graphic characters used to bound elements of the output in a clear and readable form.

2.2.1.1 *Graphic characters*

Graphic characters are used to improve readability of output.

2.2.1.2 *Format effector*

A format effector is used to format output in a suitable manner. Certain format effectors are specifically incorporated in the output definition given in § 3, but where the format effector element is shown any of the format effectors specified for MML can be used. No syntax diagram is shown.

2.2.2 *Source identifier*

A source identifier indicates the physical area in which an output was generated.

2.2.3 *Calendar date*

The output of the date in the header is based on the International Standard (ISO 2014) [1] for the writing of calendar dates in all-numeric form. The calendar date shall be written in the following order: year, month, day. The calendar date shall consist of a two decimal digit or four decimal digit year, a two decimal digit month, and a two decimal digit day of the month. The allowable characters between year and month and between month and day are hyphen or space.

Examples:

The 4th October 1979 shall be written in one of the following ways:

a) 19791004;

b) 1979-10-04;

c) 1979 10 04;

d) 791004;

e) 79-10-04;

f) 79 10 04.

The calendar date in input should preferably have a layout similar to that in output.

2.2.4 *Time of day*

The output of the time in the header is based on the International Standard (ISO 3307) [2]. However, in MML the output of a decimal fraction of hours, minutes, or seconds is not utilized in the header.

Time representations are based upon the 24-hour timekeeping system. The sequencing of time elements shall be from high order to low order (left to right): hours, minutes, seconds. The hour shall be represented by a two-digit decimal number ranging from 00 up to and including 23. The minute shall be represented by a two-digit decimal number ranging from 00 up to and including 59. The second shall be represented by a two digit decimal number ranging from 00 up to and including 59.

Examples:

Hours, minutes	1225	or 12:25
Hours, minutes, seconds	122501	or 12:25:01

2.2.5 *Additional header information*

Additional header information is general information which has no relation to the function of the output, e.g.:

- sequence number,
- processor number,
- output device,
- day of the week.

2.3 *Alarm statement*

The alarm statement may give information of a general class such as the degree of alarm or the source of alarm.

2.3.1 *Variable text*

Variable text is a set of information units which contains information unique to the event which caused the output.

2.4 *Additional information*

Additional information is general information related to the output, e.g.:

- type of output e.g., maintenance, statistics. This is not the same as identification of output, (see § 2.6),
- output recipient identification.

2.5 *Command reference*

A command reference supplies a command sequence number when needed in output outside dialogue as a reference to a previous input. In addition to the command sequence number it may also include clarifying text. It also may appear in dialogue procedures (see Recommendation Z.317).

2.5.1 *Clarifying text*

Clarifying text is a set of information units used to make the purpose and contents of the output more clear to the reader. Several clarifying texts could appear in an output.

2.6 *Identification of output*

Identification of output provides a unique identity for an output in a system's repertoire of outputs. Therefore, it could be used as a reference to the explanation of the output in a manual.

2.7 *Text block*

A text block is any combination of clarifying texts, variable texts, parameter name defined parameters and/or tables which gives information wherever it is needed or requested. For VDT applications this may be a displayed form.

2.8 *Table*

A table is an ordered presentation of interrelated information.

Clarifying text within a table can be used as labels to each column contained within the table. Where a table name or additional information associated with the table is required the clarifying text appearing at the beginning of the table in the syntax diagram of § 3.8 could be used.

When parameter name defined parameters are used to label columns each parameter should be complete, i.e. contain a parameter value (see Recommendation Z.315).

2.8.1 *New line*

New line is a character combination necessary to reset an output device to the beginning of a new line. It is recognized that the character combination is device dependent but can contain the characters CR (carriage return) and LF (line feed). No syntax diagram is shown.

2.9 *End of output*

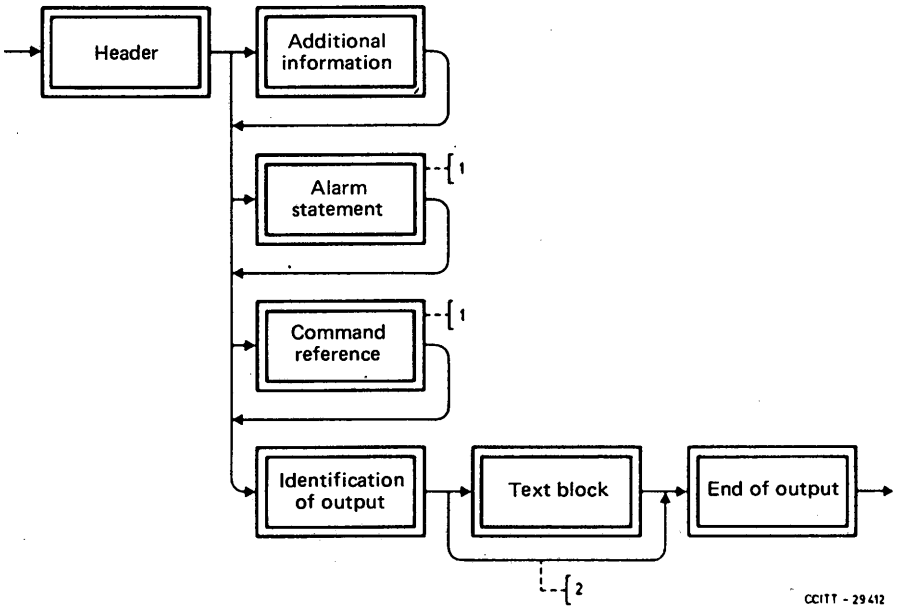
An end of output is an indication that an output is finished.

2.10 *Comments in output*

The purpose of a comment in output is as for clarifying text (see § 2.5.1) with the exception that the syntax is as for comment in input so that it may be discarded during a subsequent re-input. No syntax diagram is shown.

3 Definition of the output language syntax in diagrams

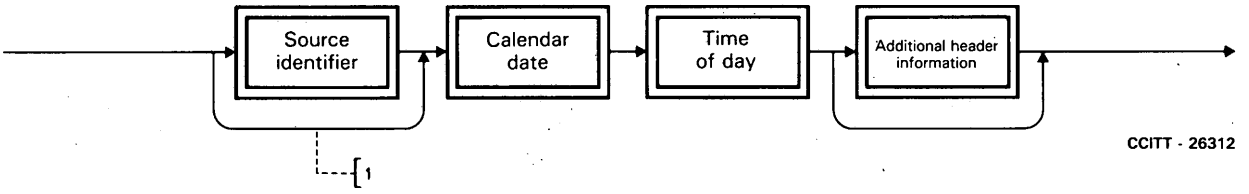
3.1 *Output outside dialogue*



1) Command reference and alarm statement could appear in the same output, e.g. if a control system unit is taken out of service by means of a command.

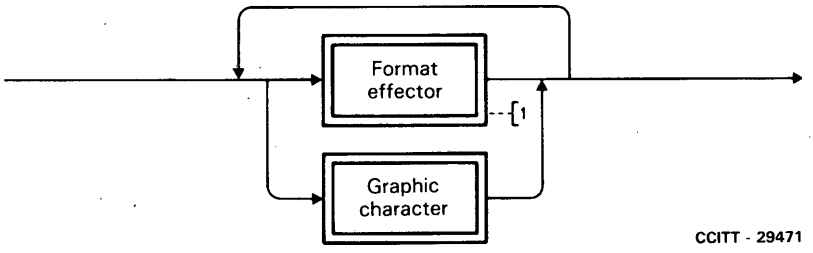
2) This by-pass can be taken only when the identification of output contains sufficient information.

3.2 *Header*



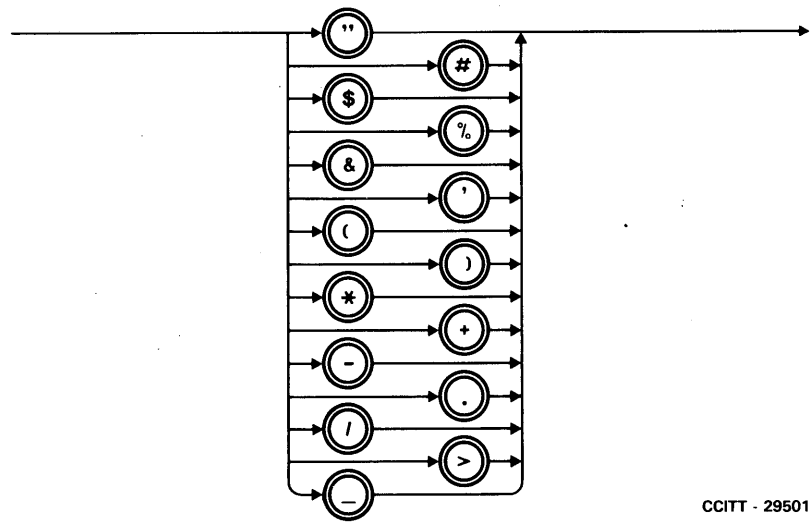
1) Source identifier may be omitted where there is only one source producing outputs.

3.2.1 *Layout option*

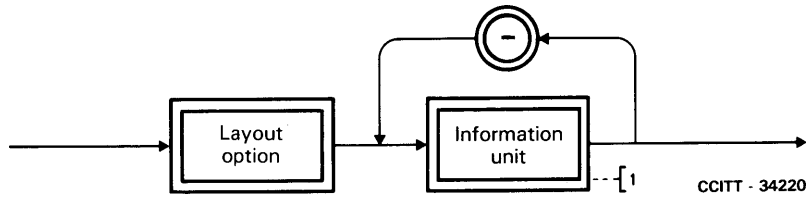


1) Not further expanded in diagram form.

3.2.1.1 *Graphic character*

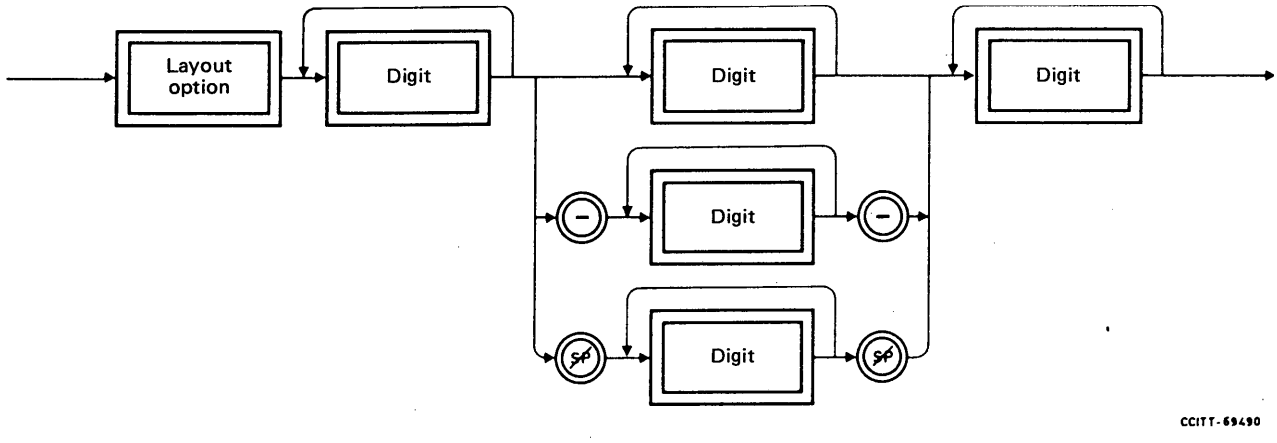


3.2.2 *Source identifier*

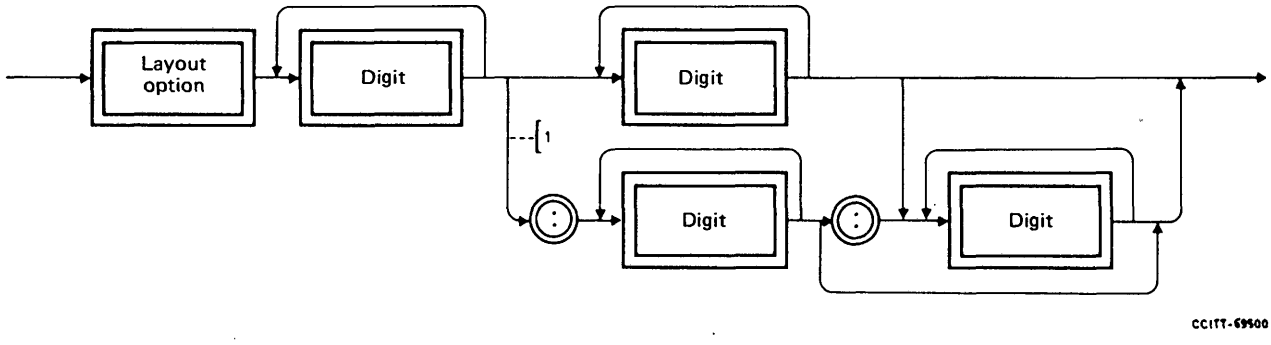


1) See Recommendation Z.315.

3.2.3 *Calendar date*



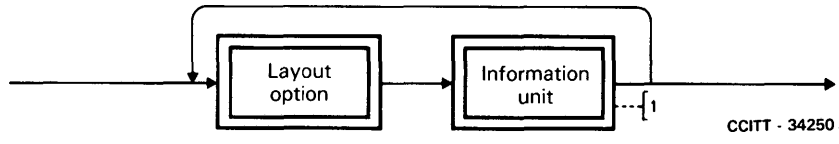
3.2.4 *Time of day*



1) a) If required to facilitate visual human understanding of output, a : (colon) may be used to separate hours, minutes and seconds (refer to [2]).

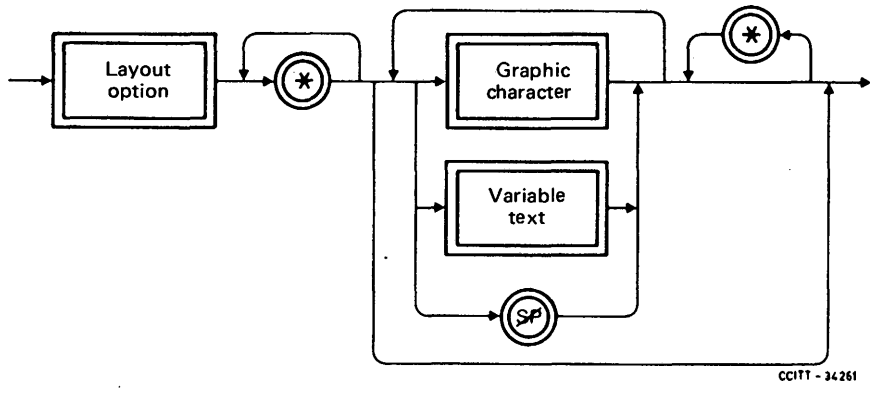
b) This use of the : (colon) is not allowed in input since the character is used as a separator between blocks of parameters.

3.2.5 *Additional header information*

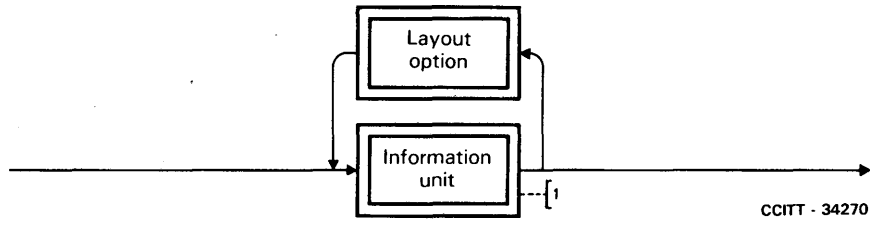


1) See Recommendation Z.315.

3.3 *Alarm statement*

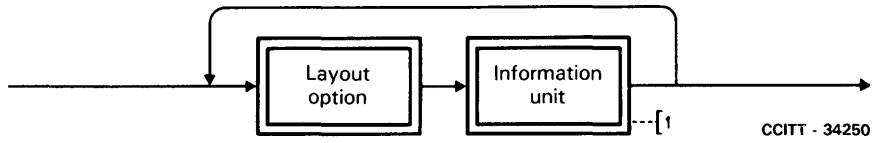


3.3.1 *Variable text*



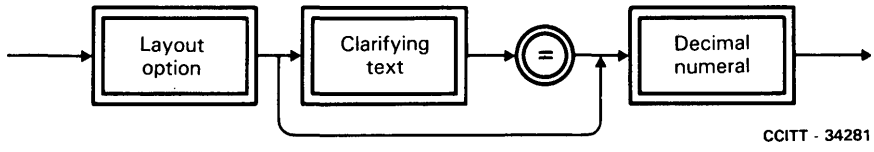
1) See Recommendation Z.315.

3.4 *Additional information*

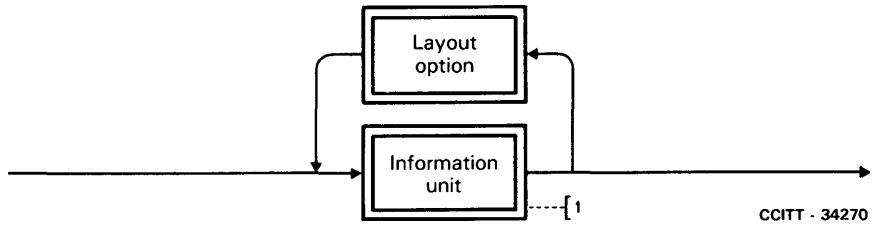


1) See Recommendation Z.315.

3.5 *Command reference*

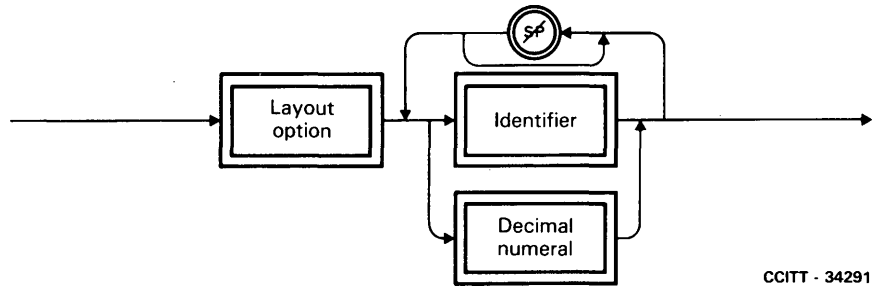


3.5.1 *Clarifying text*

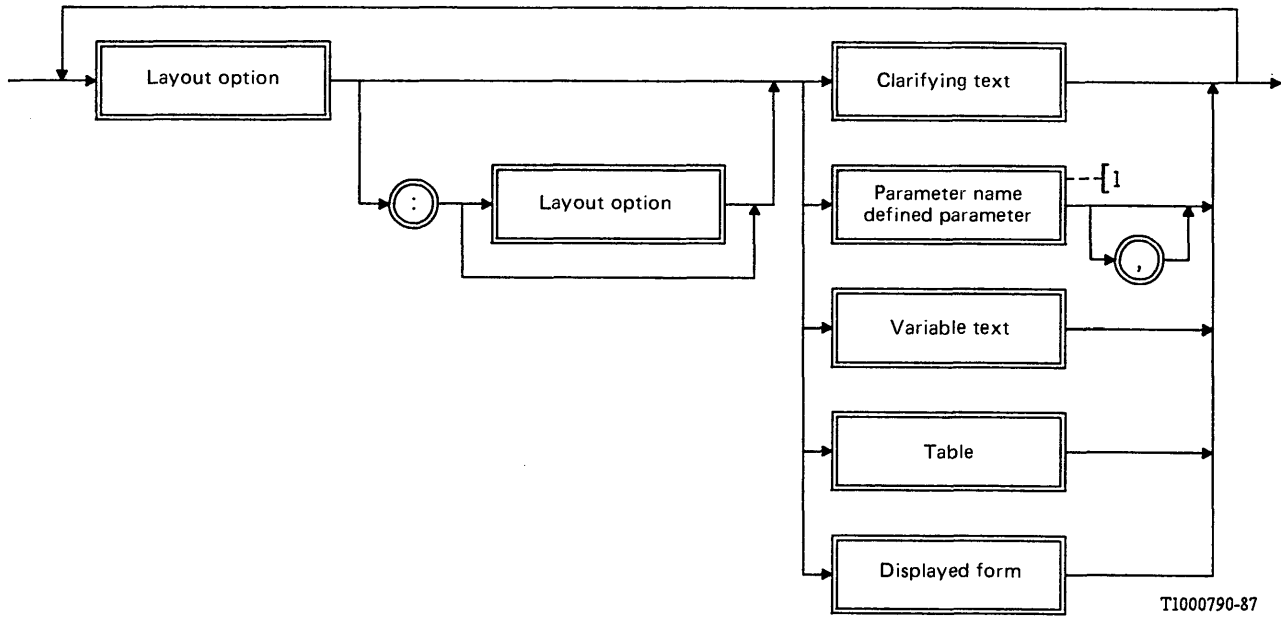


1) See Recommendation Z.315.

3.6 *Identification of output*

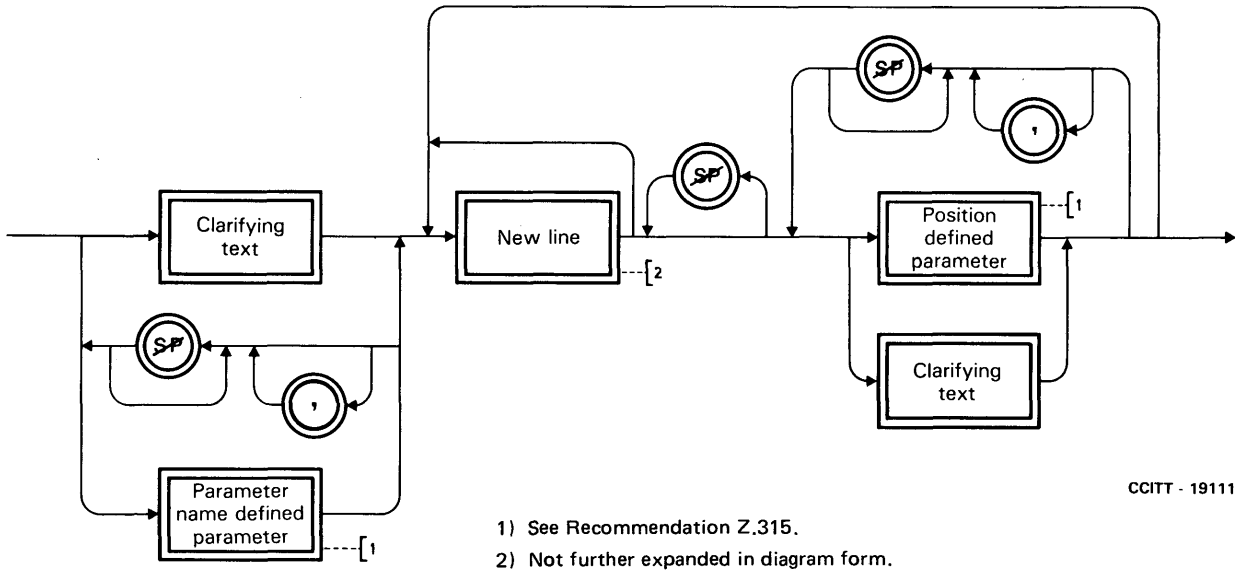


3.7 *Text block*



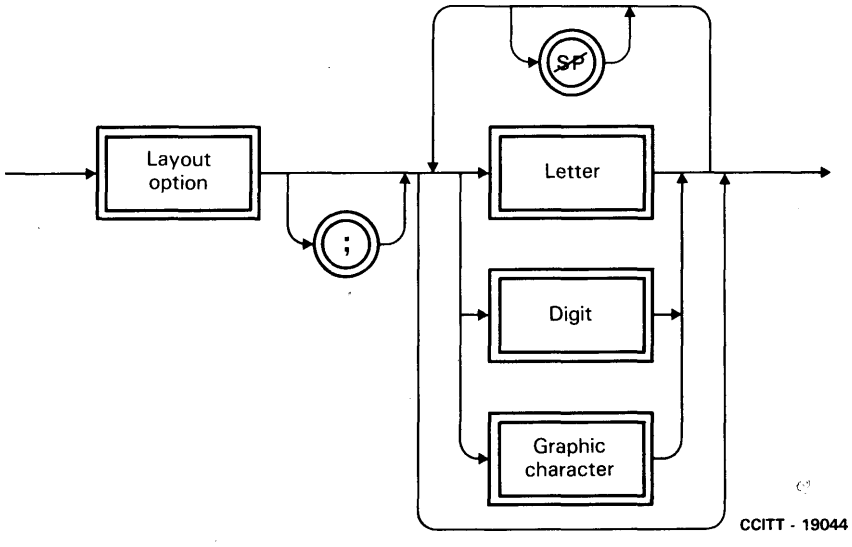
1) See Recommendation Z.315.

3.8 *Table*



1) See Recommendation Z.315.
2) Not further expanded in diagram form.

3.9 *End of output*



References

[1] *Writing of Calendar Dates in All-Numeric Form*, ISO Standard 2014-1976.

[2] *Information Interchange – Representation of Time of the Day*, ISO Standard 3307-1975.

Recommendation Z.317

MAN-MACHINE DIALOGUE PROCEDURES

1 General

Man-machine communication comprises two types of information interchange, namely *dialogue* and *output outside dialogue*; they occur sequentially and in no particular order. Output outside dialogue is fully defined in Recommendation Z.316.

Dialogue is that part of man-machine communication initiated and, normally, terminated by the user. It is accomplished by means of the dialogue procedures described in this Recommendation. In the text, the terms "dialogue" and "dialogue procedure" are used interchangeably.

The text in § 2 describes the dialogue procedure, the syntax diagrams of which are given in § 3 in sub-divisions having numbers corresponding to those used in § 2.

A systematic analysis of possible errors made by users is not considered. Diagrams mainly refer to correctly given commands and only obvious error situations are considered. It is recognized that the diagrams are not exhaustive and some of them might be modified when error recovery procedures have been completely considered.

2 Definition of the dialogue procedure

2.1 *Overview of the* dialogue procedure

A dialogue is opened by a procedure prologue. The procedure prologue contains the various preparations which must be performed before commands can be initiated. It may include a header from the system. Following the procedure prologue a destination prologue can precede one or more interactive operating sequences. The dialogue can be terminated by a procedure epilogue.

2.2 procedure prologue

The procedure prologue may consist of three parts given in the following order:

- the request, which is an action to activate the man-machine terminal and the system;
- the identification of the user. The identification of the user is optional. Identification may be bypassed under special conditions, for example system initialization. In situations where no identification procedure is used, then it must be possible to allow access only for certain periods per day, e.g., office hours;
- a header, which is given from the system and contains the exchange identification, information relating to date and time, etc. Headers can be optional for a system or within a system for certain terminals.

The procedure prologue is intended to be executed only once at the beginning of a dialogue. The procedure prologue is followed by a ready indication inviting a destination prologue or an interactive operating sequence.

The request, the identification of the user and the header are defined in the following paragraphs.

2.2.1 request

The request is a manual action to activate the terminal and the system or to cause an interrupt. The composition of the request is highly dependent on the type of terminal and implementation.

The request can consist of keying the break key or actuating a control switch, power on, etc. and/or keying a sequence of characters on the keyboard.

2.2.2 identification procedure

The identification procedure is used to identify the user to the system. The identification procedure may involve the use of identity cards which provide secure access to the system.

After a user has been identified to the system, different authorization levels may be applied that restrict access to groups of commands depending on security or functional classification.

The identification procedure (see figure 3.2.2/Z.317) is flexible, with many options, but the following guidelines apply:

- if an identity card is used, it should always be preceded or followed by a password;
- for security reasons, it might be required to suppress all response from the system to the identification procedures;
- after a number of consecutive attempts some appropriate action is needed. For example: generate an alarm, or temporarily block access to the system from that terminal.

2.2.2.1 ready indication

The ready indication indicates that the direction of the dialogue has changed and that the system is waiting for information to be given at the terminal. The ready indication is defined as the character < (less than sign) optionally preceded by the appropriate format effectors. The < (less than sign) character is not necessarily required in extended MML (Recommendations Z.321-Z.323), as the information that the terminal is ready for input can be given by cursor position, or additional information contained somewhere in the menu or form.

2.2.3 header

The header (see Recommendation Z.316) is output by the system at the end of the procedure prologue.

2.3 destination prologue

The destination prologue consists of a destination identifier terminated by the separator > (greater than sign) so as to distinguish it from a command.

The destination identifier indicates the physical area where the command is to be mainly processed, e.g., exchange identification, processor number. It consists of one or more information units separated by - (hyphen). The destination could also be defined by a parameter in the command.

The destination identifier may be followed by a header to indicate that a selected destination is allowed, available and ready or alternatively by a rejection output to indicate the converse.

2.4 **procedure epilogue**

The procedure epilogue is used to terminate the dialogue procedure. The composition of the procedure epilogue is highly dependent on the type of terminal and implementation. The procedure epilogue can consist of actuating a control switch, power off, etc. and/or keying a sequence of characters on the keyboard and/or the output of end of dialogue from the system.

2.5 **interactive operating sequence**

The interactive operating sequence may consist of a single command entry sequence terminated by an optional end statement or of a series of command entry sequences or special actions. The latter occurs when, as a result of partial execution of a function, the system requests the supply of further information in the form of special actions or further commands for which human judgement and/or decision is required.

2.5.1 **command entry sequence**

A command entry sequence contains a single command code, together with an alternating sequence of one or more parameter blocks and an appropriate number of executions.

Any interactive operating sequence may be stopped prematurely by the user with the entry of a particular command entry sequence. The latter could consist of a certain command which is independent of any interactive operating sequence, e.g., EXIT, etc.

2.5.2 *Manual response*

Special actions can include manual responses, such as the actuation of keys on terminals or switchframes and the replacement of equipment.

2.5.3 *Interaction request output*

The system generates an interaction request output in order to obtain further actions.

2.5.4 **end statement**

An end statement is an indication that an operating sequence has finished.

2.6 *Direct parameter input*

Only one method of inputting parameters is dealt with in direct parameter input. For other methods refer to Recommendations Z.321 to Z.323.

Direct parameter input consists of an optional parameter block entry sequence preceded by the separator : (colon). The none or more parameter blocks are to be terminated by the execution character ; (semicolon) or by the continuation character ! (exclamation mark) to initiate the required functions which will result in a response output.

If terminated by an execution character and responded by an acceptance or rejection output, the system concludes the direct parameter input. If terminated by a continuation character and responded by an acceptance or rejection output, the system is required to return a parameter block request indication that functions as an indication to proceed with the input of the next block or blocks of parameters. If responded by a request output the system is required to return a parameter block request indication that functions as an invitation for entering either an updated part of the current block of parameters (e.g., a parameter that was erroneously input) or an expansion of the current block of parameters, dependent on the contents of the request output. Following the parameter block request indication, the command entry sequence can be abandoned by invoking the delete command function.

The parameters are input in accordance with the parameter block entry sequence.

2.6.1 *Parameter block entry sequence*

The parameter block entry sequence is used to input a block of parameters. All parameters are entered according to the input syntax. The entry of the parameters may be done directly without help from the system as described in Recommendation Z.315, or assistance from the system may be requested by calling the prompting facility. Prompting helps in providing a correct input by the system giving guidance on the next input requirement.

The output given by the prompting facility can be either of the following:

a) Guidance output followed by a ? (question mark). The guidance may apply to the complete block of parameters, to that part of the block of parameters that is still to be input or to the single parameter next to be input. Moreover it may contain an indication that the input supplied is sufficient and that an execution order may be given. Guidance can be requested anywhere in the parameter block entry sequence.

b) Parameter name output followed by an = (equal sign). The parameter name applies to the parameter value next to be input.

It is the objective of the parameter name output or guidance output to assist the user in giving correct input required by the system for the current command. In both cases the system may verify input received – if possible – and prompt with enough information to enable input to continue.

What kind of prompting output is given is dependent on the prompting facilities supported by the system involved and – if more than one facility is supported – on the place of the request for prompting.

These recommendations address prompting on request of the user. Unsolicited system directed prompting is also possible but is not covered by these recommendations.

Following "parameter name output", a default value for the parameter cannot be implied by simply omitting the value. A specific "default indicator" must be given. If, however, a further ? (question mark) is input, the system will give guidance output, and default by omission may then be possible.

2.6.2 *Parameter block request indication*

The parameter block request indication consists of a : (colon) optionally preceded by the appropriate format effectors and/or the appropriate command code.

2.7 *Response output*

Response output covers all types of output conveying information about the state of an input. Types of response output are acceptance output, rejection output and request output.

A list of categories of each type of response output is given below. Each category is identified by means of the status of the requested action or by means of the error introduced by the user. The title of each category is not meant to be interpreted as the text to be associated with each response output. Additional categories may be created, e.g., by dividing into several parts any one of the categories listed below.

2.7.1 *Acceptance output*

Acceptance output is an indication that an input to the system is syntactically correct and complete and that the appropriate system actions will be initiated, or have already been carried out. In the latter case, this indication may take the form of the result of the actual action.

Category of acceptance output	*Description*
COMMAND EXECUTED	The input command was correct and the requested action(s) was successfully performed. The execution of some commands may produce a result to be output immediately after the command has been input. In this case, the result itself may act as the acceptance output.
COMMAND ACCEPTED	The input command was correct and the requested action(s) was accepted. This action(s) is either in progress or has been scheduled to be performed. Subsequent outputs related to this requested action may follow later.

2.7.2 *Rejection output*

Rejection output is an indication by the system that the input received is not valid and will not be acted upon, nor can correction be applied, e.g., when the system determines that the user is not authorized to request the action required by the command.

Category of rejection output	*Description*
UNACCEPTABLE COMMAND	The command form is valid but the requested action conflicts with the current system or equipment status, e.g., an attempt to restore an in-service unit.
NO SYSTEM RESOURCES	The requested action cannot be executed now due to unavailable system resources such as system overload, excessive queue lengths, busy programs, etc. The command may be entered again later.
TRANSMISSION ERROR	A transmission error occurred in the input and the system will not accept the command.
SYSTEM ACCESS UNAVAILABLE	Input/output access to the system is currently unavailable.
GENERAL ERROR	Any rejection that cannot be placed in one of the more specific rejection output categories.
INVALID PASSWORD	The input password is unknown to the system or has been input from an improper terminal.
ILLEGAL COMMAND	The input command cannot be requested under the current password or from the terminal from which it has been requested.
INVALID SEQUENCE	In an interactive operating sequence a command has been entered in the wrong sequence.
UNKNOWN COMMAND CODE	The input command is not recognized by the system.
TIME OUT ERROR #1	The next input character has not been received in time for processing and the command has been aborted.
INVALID COMMAND CODE SEPARATOR	The command code contains an invalid separator.
INVALID COMMAND CODE IDENTIFIER	The command code contains an invalid identifier.

2.7.3 *Request output*

Request output is an output message which requests further input action, e.g., to correct an erroneous parameter.

Category of request output	*Description*
INVALID SEPARATOR	The wrong input character has been used as a separator.
INVALID INDICATOR	The wrong input character has been used as an indicator.
INVALID PARAMETER NAME	A parameter name not associated with this command has been input.
EXTRA PARAMETERS	Too many parameters have been entered or a parameter has been entered in a command not requiring parameters.
MISSING PARAMETER	One or more parameters required by the command have not been entered.
INCONSISTENT PARAMETER	The set of parameters in a command does not form a valid set, or the parameters received at an intermediate point are not a valid subset.
MISSING DATA	One or more information units of a parameter argument have been omitted.
INCONSISTENT DATA	One or more parameter arguments are inconsistent with arguments associated with other parameters, or with the presence (absence) of other parameters in the command, or with data already in the system, although each could be individually valid.

INVALID INFORMATION GROUPING	The type of information grouping used in the input of the parameter value is not valid.
RANGE. ERROR	The value(s) assigned to a parameter is out of the range of the allowed values.
INVALID INFORMATION UNIT	The information unit(s) introduced to specify the value(s) of a parameter does not match with the syntactic element requested for the information unit(s).

2.7.4 *Miscellaneous output*

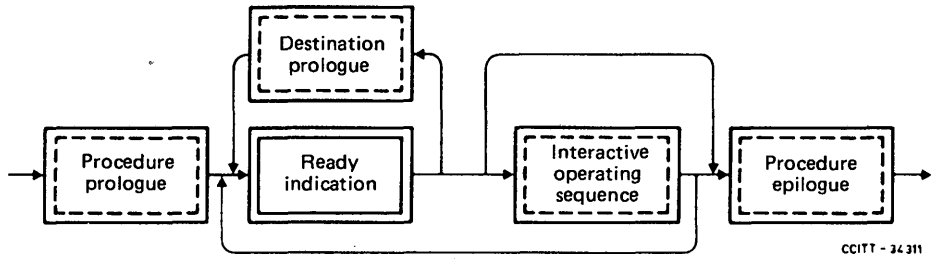
A category of output that does not belong to one of the types above is that given when the dialogue is closed on the initiative of the system.

Category of output	*Description*
TIME OUT ERROR #2	The next input after the completion of a command has not been received in time and the dialogue has been aborted.

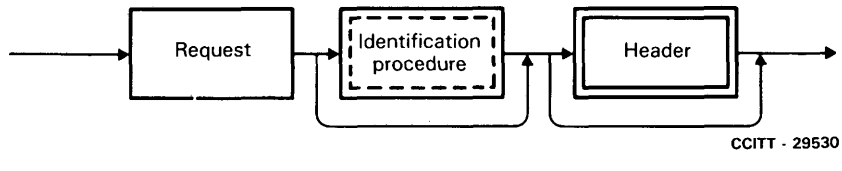
3 Definition of the dialogue procedure syntax in diagrams

Recommendations Z.315 and Z.316 describe the input and output syntactic elements used, but not defined, in this Recommendation.

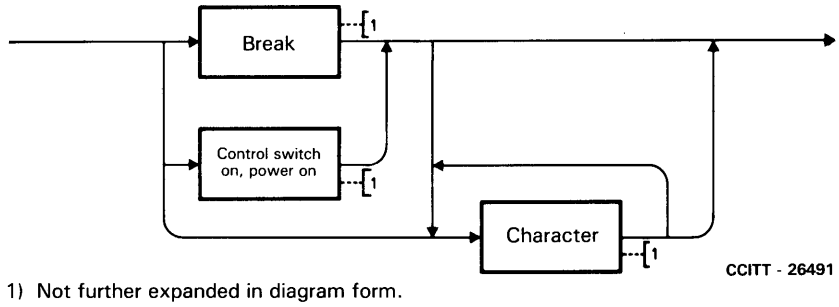
3.1 *Dialogue procedure*



3.2 *Procedure prologue*

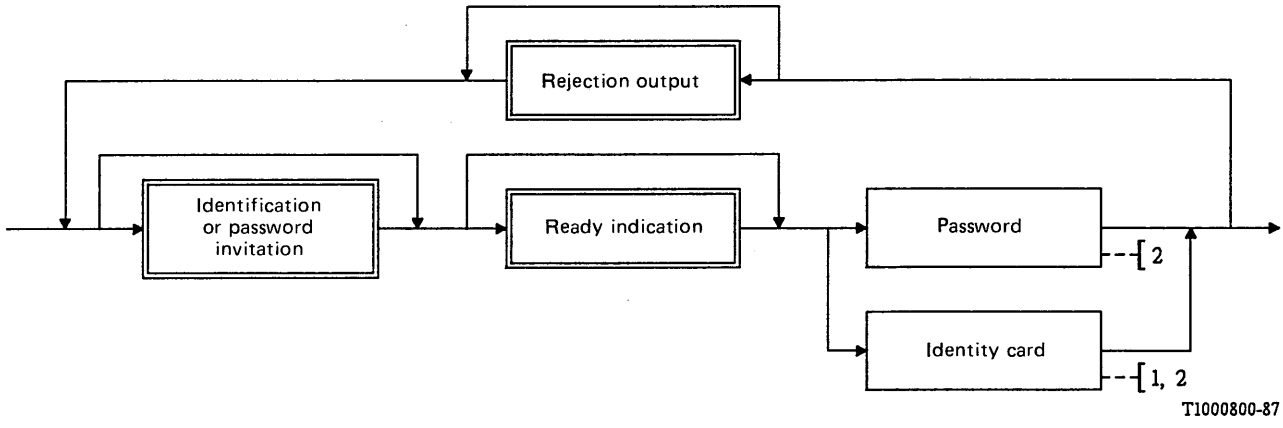


3.2.1 *Request*



1) Not further expanded in diagram form.

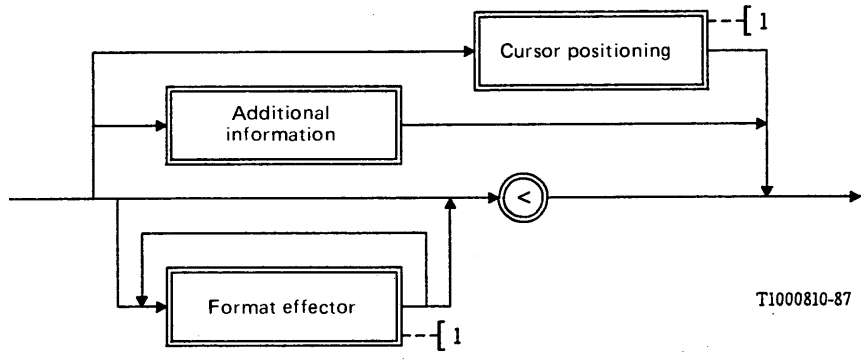
3.2.2 *Identification procedure*



1) Not further expanded in diagram form.

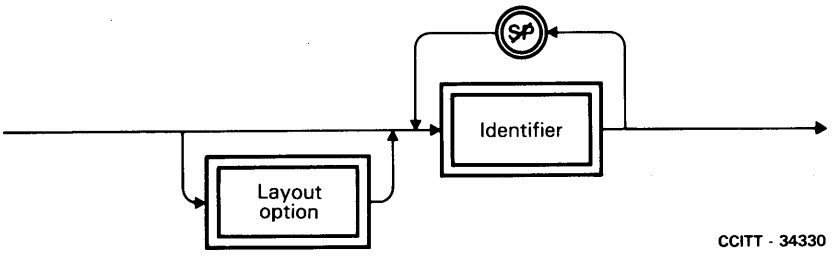
2) If an identity card is used, it should always be preceded or followed by a password.

3.2.2.1 *Ready indication*

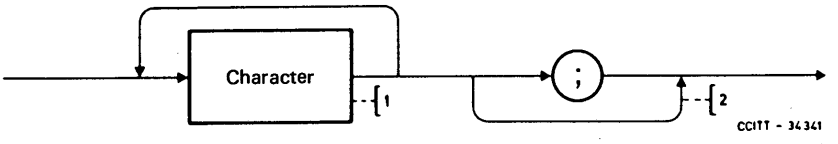


1) Not further expanded in diagram form.

3.2.2.2 *Identification or password invitation*



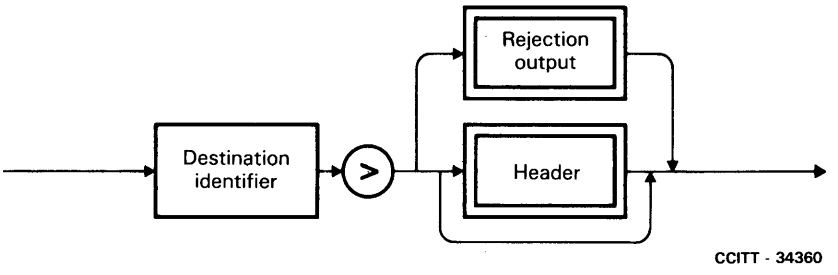
3.2.2.3 *Password*



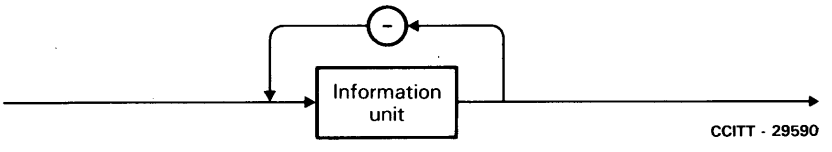
1) Not further expanded in diagram form.

2) If an explicit MML indicator is used to terminate the input, it is recommended to be the ; (semicolon). On the other hand the bypass reflects that other mechanisms to terminate the input are available, e.g. an implicit length of a password.

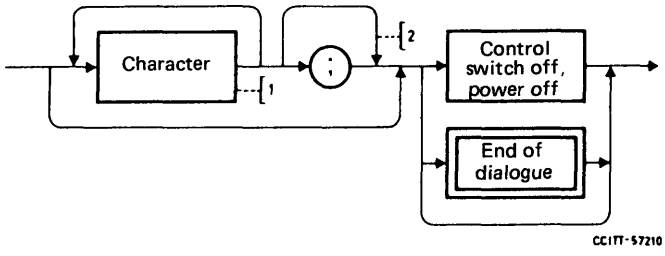
3.3 *Destination prologue*



3.3.1 *Destination identifier*



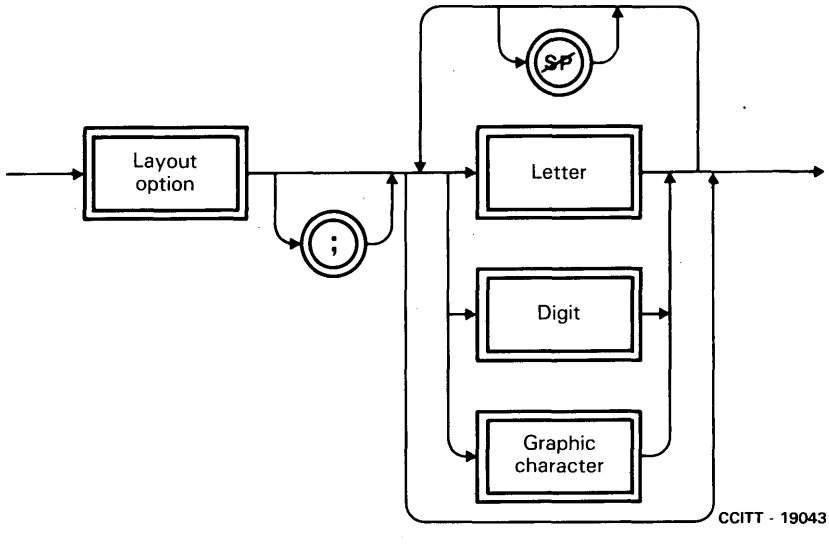
3.4 *Procedure epilogue*



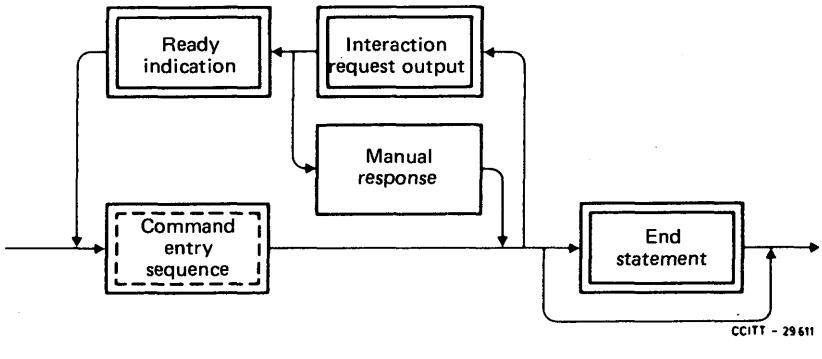
1) Not further expanded in diagram form.

2) If an explicit MML indicator is used to terminate the input it is recommended to be the ; (semicolon). On the other hand the bypass reflects that other mechanisms to terminate the input are available, e.g. a unique set of characters such as "OFF", "BYE".

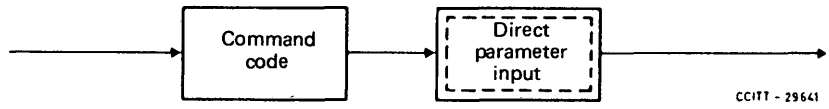
3.4.1 *End of dialogue*



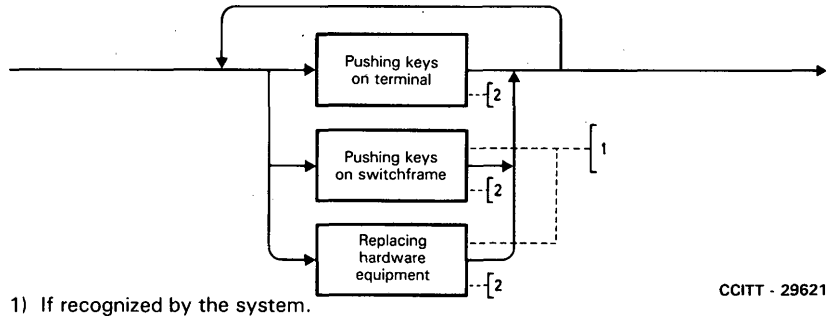
3.5 *Interactive operating sequence*



3.5.1 *Command entry sequence*

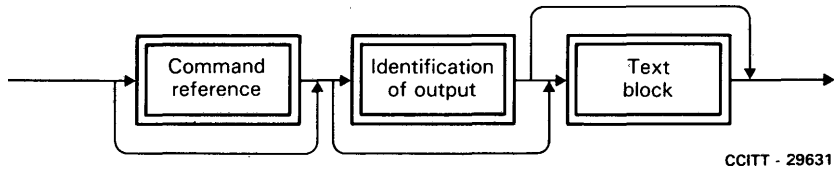


3.5.2 *Manual response*

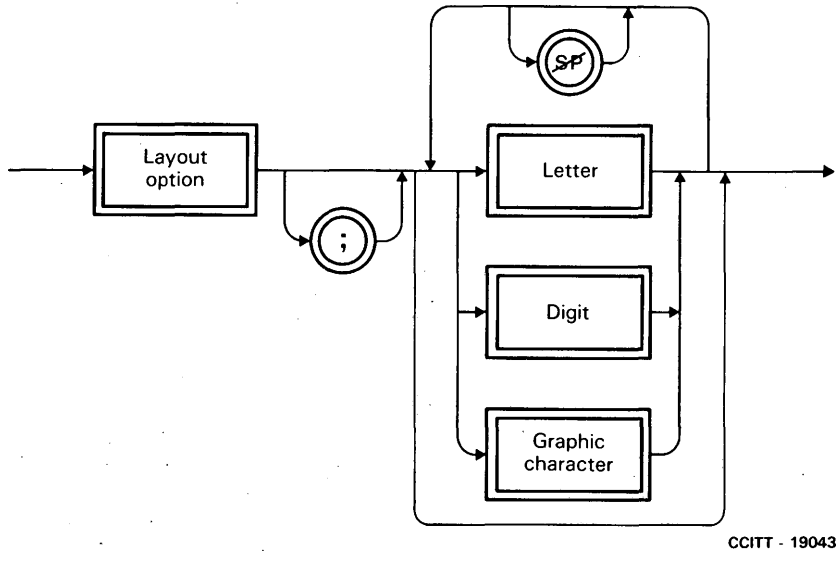


1) If recognized by the system.
2) Not further expanded in diagram form.

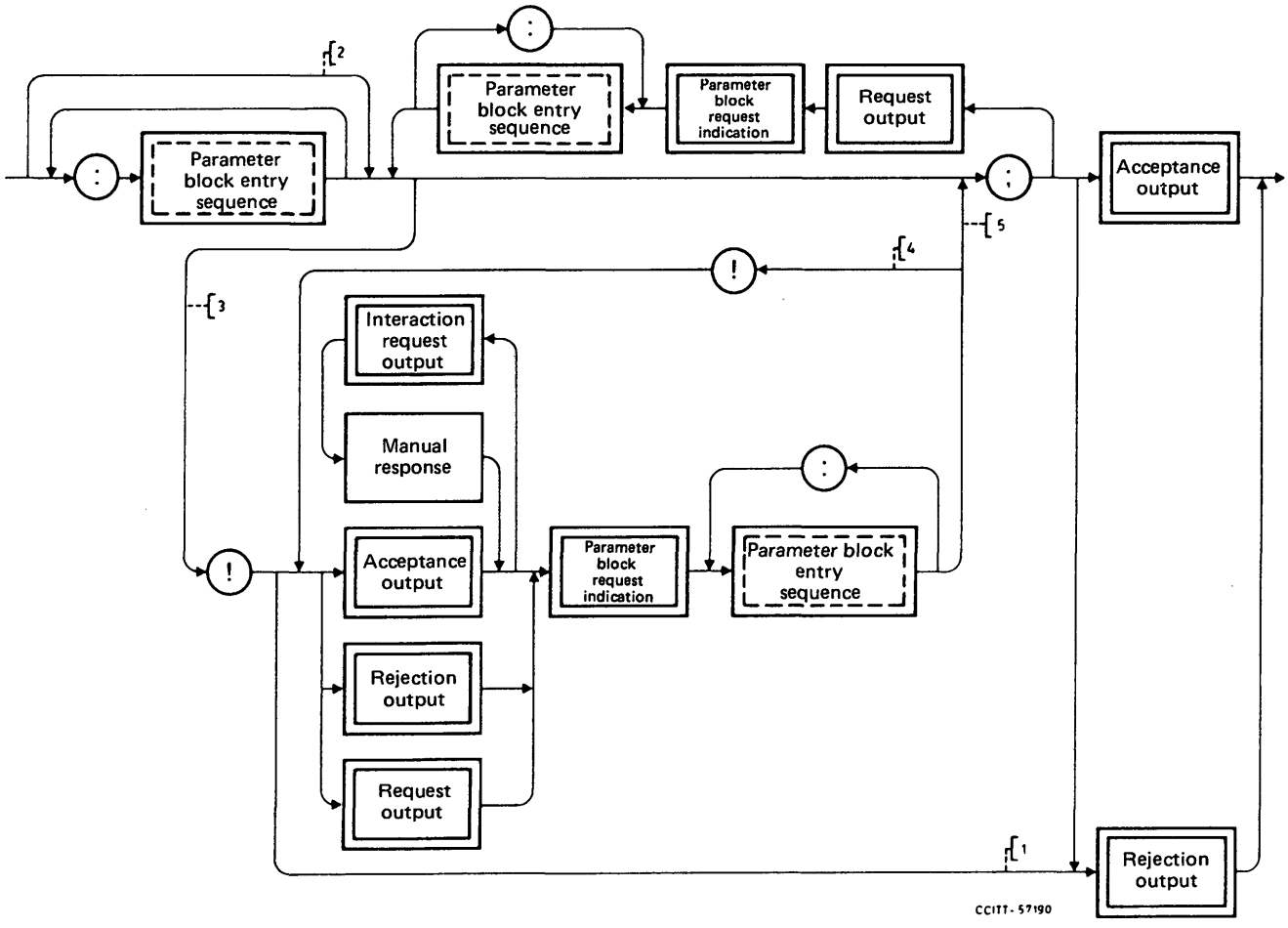
3.5.3 *Interaction request output*



3.5.4 *End statement*

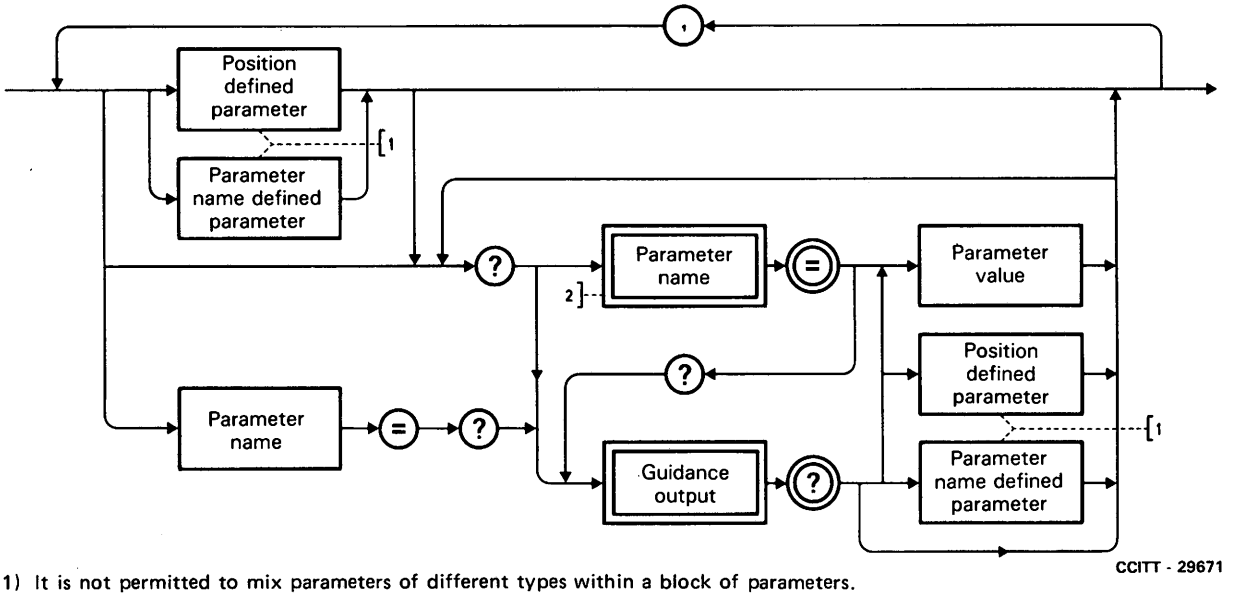


3.6 *Direct parameter input*



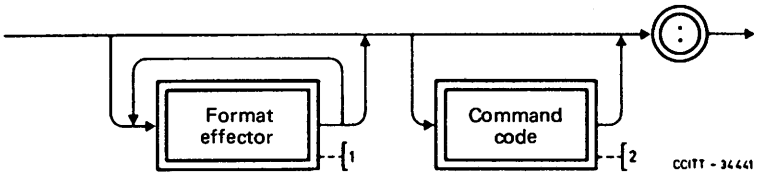
1) Only if command code is not valid.
2) Command without parameters or with default parameters only.
3) First command of a continuation series.
4) Subsequent command of a continuation series.
5) Last command of a continuation series.

3.6.1 *Parameter block entry sequence*



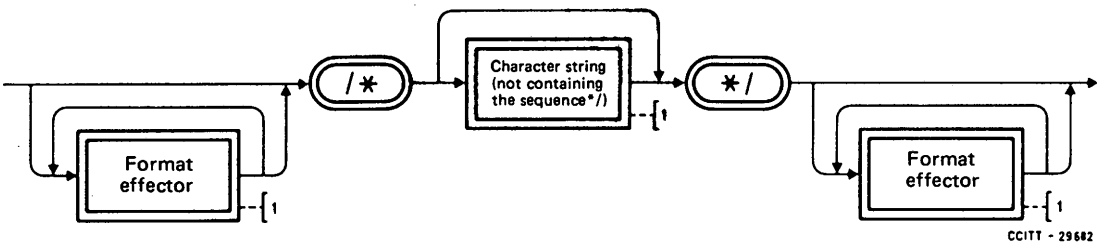
1) It is not permitted to mix parameters of different types within a block of parameters.
2) See Recommendation Z.315.

3.6.2 *Parameter block request indication*



1) Not further expanded in diagram form.
2) See Recommendation Z.315.

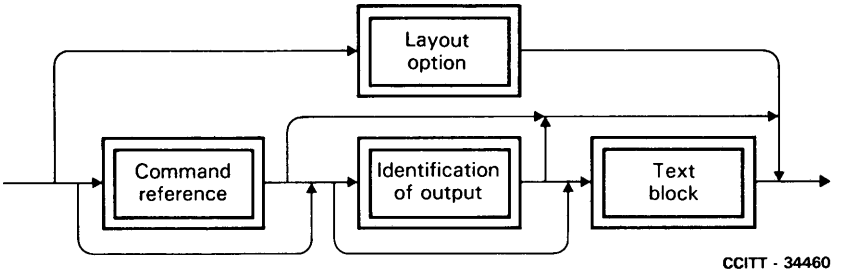
3.6.3 *Guidance output*



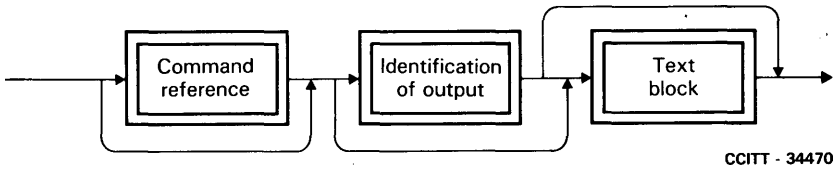
1) Not further expanded in diagram form.

3.7 *Response output*

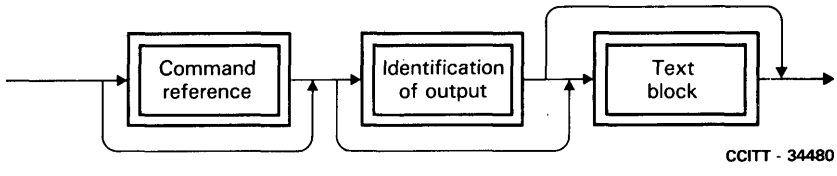
3.7.1 *Acceptance output*



3.7.2 *Rejection output*



3.7.3 *Request output*



4 Input/output management

4.1 *General*

The question of input/output management is highly hardware and system dependent. Input/output management strategies should be provided to:

- solve any conflict of output outside dialogue directed to an input/output (I/O) device involved in a dialogue procedure;
- solve any conflict of more than one output outside dialogue competing for the same I/O device;
- permit the user to perform a dialogue at any time.

4.2 *Priorities of output*

The priority of an output outside dialogue will determine the behaviour of the output in relation to a dialogue procedure and in relation to other outputs. System crash messages and those outputs that occur after a dangerous situation, implying an immediate recovery procedure such as system reload, are not governed by the following input/output management procedures but may be output at any time.

The priority of an output outside dialogue is the property of the output and dictates the sequence of the output. When several outputs are competing for the use of the same I/O device, the output with the highest priority is output first. Outputs of the same priority are output on a first come first served basis. From an input/output management point of view there shall be two classes of priority for output outside dialogue: high, low.

Lengthy outputs shall be divided into convenient units. Interruptions of output shall only occur at the end of an output unit. A suitable dimension for a unit of output shall be sufficient to allow the output of a meaningful message.

4.3 *Output to a device not in a dialogue procedure*

An output outside dialogue directed to an I/O device not involved in a dialogue procedure is always output, unless another output is in progress on that I/O device, in which case the current output must be completed first. These outputs may be interrupted by input (see § 4.5).

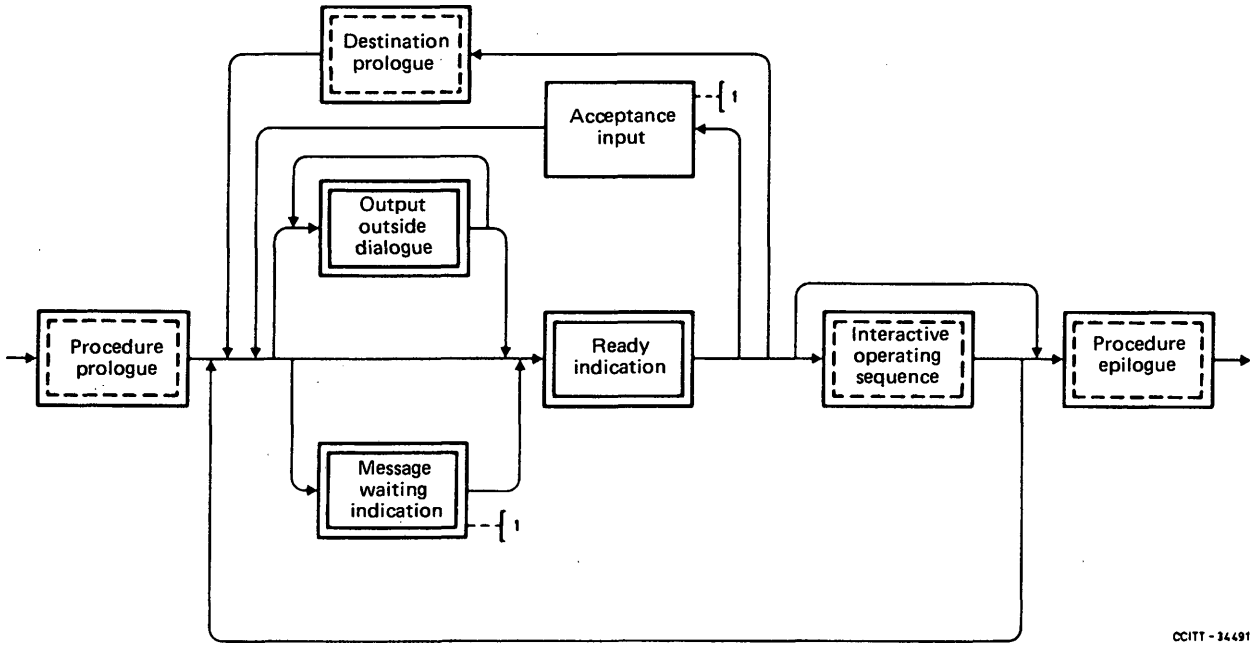
Optionally a system may choose to output the current output only up to the end of the current unit of output before outputting a waiting high priority output.

4.4 *Output to a device in a dialogue procedure*

High priority outputs, which are outputs outside dialogue, are allowed either to be announced or to interrupt the dialogue between interactive operating sequences[1]. When a high priority output is announced by means of a message waiting indication, an acceptance input can be given which will cause the waiting output to take place (see § 4.4.1 for an extended syntax diagram for output interrupting input).

Low priority outputs, which are outputs outside dialogue, are not allowed to be announced or to interrupt the dialogue and should be delayed until the end of the dialogue.

4.4.1 *Interruption in dialogue due to input/output management*



1) Not further expanded in diagram form.

4.5 *Input interrupting output*

A facility is provided to allow the interruption of an output occurring at an I/O device. However, a request, rejection or acceptance output (where it is not used as the result of the actual action) cannot be interrupted. The output may be interrupted by means of a request as defined in § 2.2.1. When the above request has been made the dialogue with the system can be started/continued.

The interrupted output may be managed by giving an instruction to resume, cancel or restart it. Alternatively, the interrupted output may be managed according to the property of the message itself, assigned at the time of message design.

When the interrupt request is given, the interrupt shall be carried out after the current unit of output.

---

1) Interruption in other places is not excluded.

5 Time-out control inside dialogue

Two particular time-outs are identified within a dialogue. The time-outs are provided to prevent lockout of outputs and/or to prove the presence of the user. The latter is used when the system has functions for procedure prologue and epilogue. In this case, two time-outs may be provided where the first one is used within any input. The second time-out is set after completion of the procedure prologue, the destination prologue, and the command entry sequence. Both time-outs are cancelled by the receipt of any input.

When the first time-out elapses, it is suggested that cancellation of the actual input should occur. When the second time-out elapses, it is suggested that the epilogue procedure should take place. Any output can take place when the first time-out has elapsed.

ANNEX A

(to Recommendation Z.317)

Use of SDL to describe MML dialogue procedures

A.1 *Introduction*

The specification and Description Language (SDL) described in the Z.100 series Recommendations can be used to describe MML dialogue procedures. This annex provides SDL examples of MML dialogue procedures from Recommendation Z.317.

A.2 *SDL description of dialogue procedures*

The SDL diagrams in figures A-1/Z.317 to A-3/Z.317 cover the main procedural aspects described in § 3 of Recommendation Z.317, excluiding the "Parameter entry sequence". Also, other aspects, such as I/O management and timing recommended in §§ 4 and 5 of Recommendation Z.317 have not been dealt with in the SDL diagrams.

The SDL diagrams have been developed with the aim of describing the MML interface. The SDL elements are:

SDL element	*Purpose*
INPUT	What the operator keys in
OUTPUT	System response
DECISION	A system decision point
ALTERNATIVE	Shows different implementation possibilities

The SDL diagrams correspond to the following figures in Recommendation Z.317:

Figure A-1/Z.317	Procedure prologue (§ 3.2) Request (§ 3.2.1) Identification procedure (§ 3.2.2)
Figure A-2/Z.317	Destination prologue (§ 3.3) Procedure epilogue (§ 3.4)
Figure A-3/Z.317	Interactive operating sequence (§ 3.5) Command entry sequence (§ 3.5.1) Direct parameter input (§ 3.6)

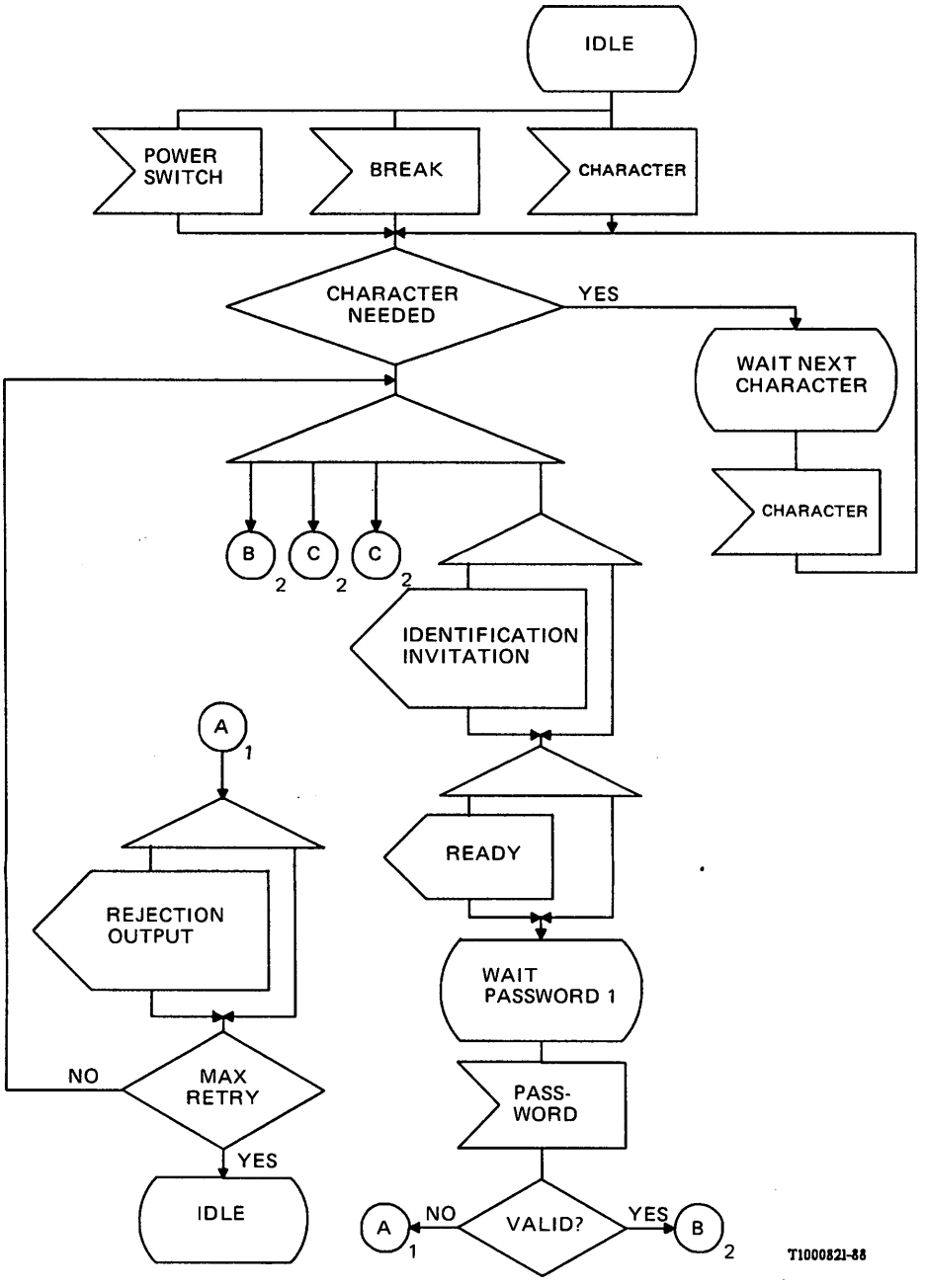


FIGURE A-1/Z.317 (1 of 2)

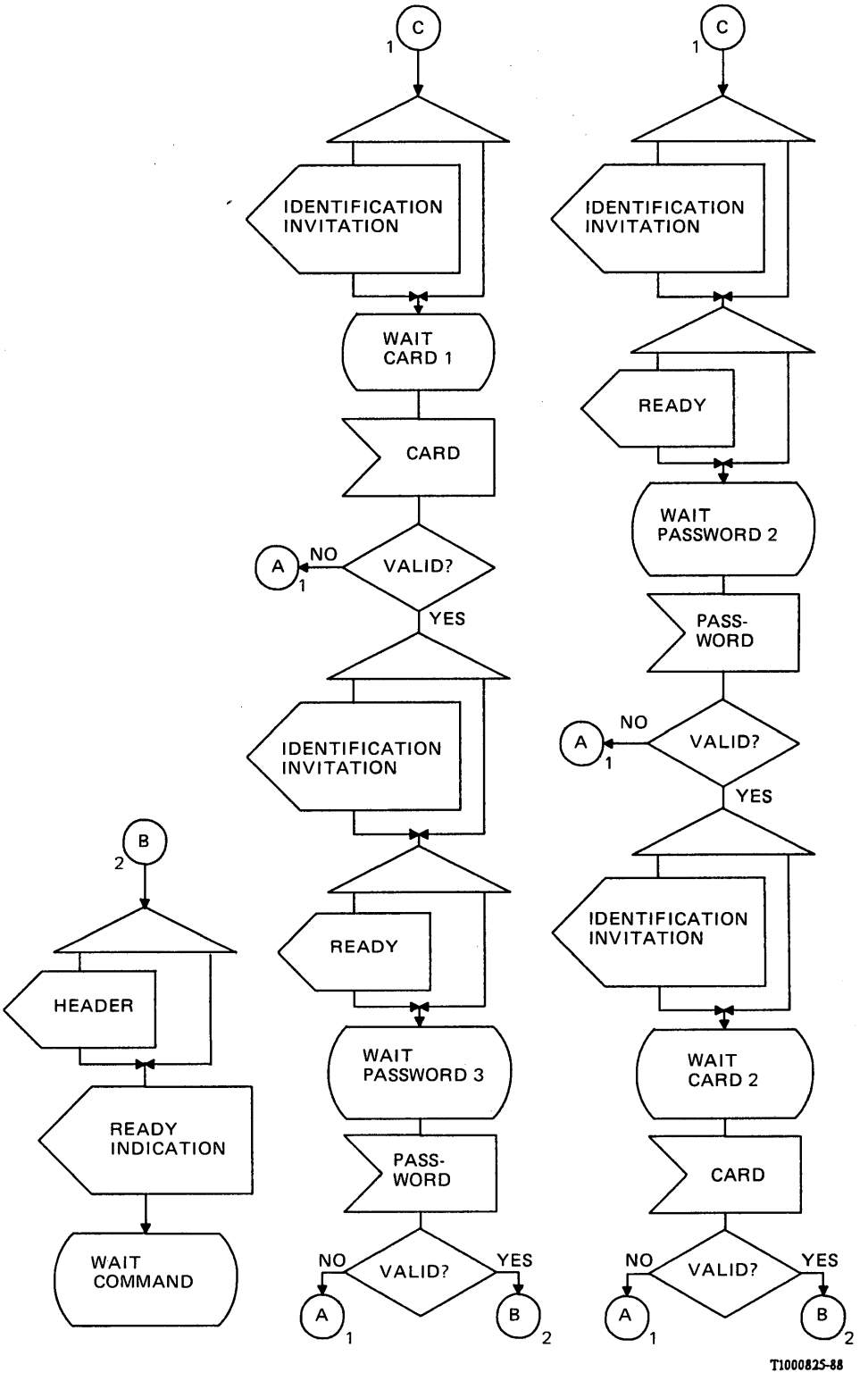


FIGURE A-1/Z.317 (2 of 2)

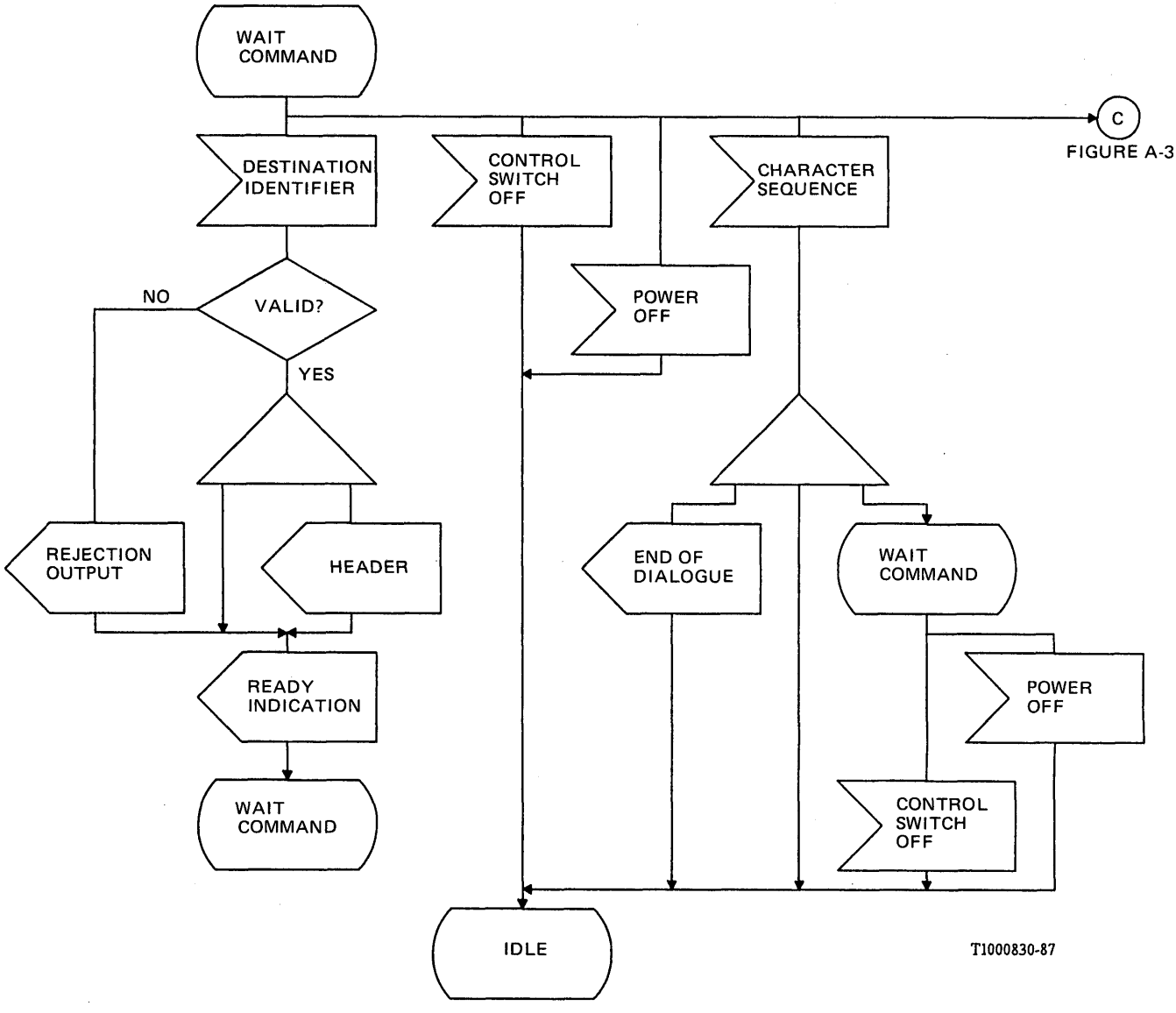


FIGURE A-2/Z.317

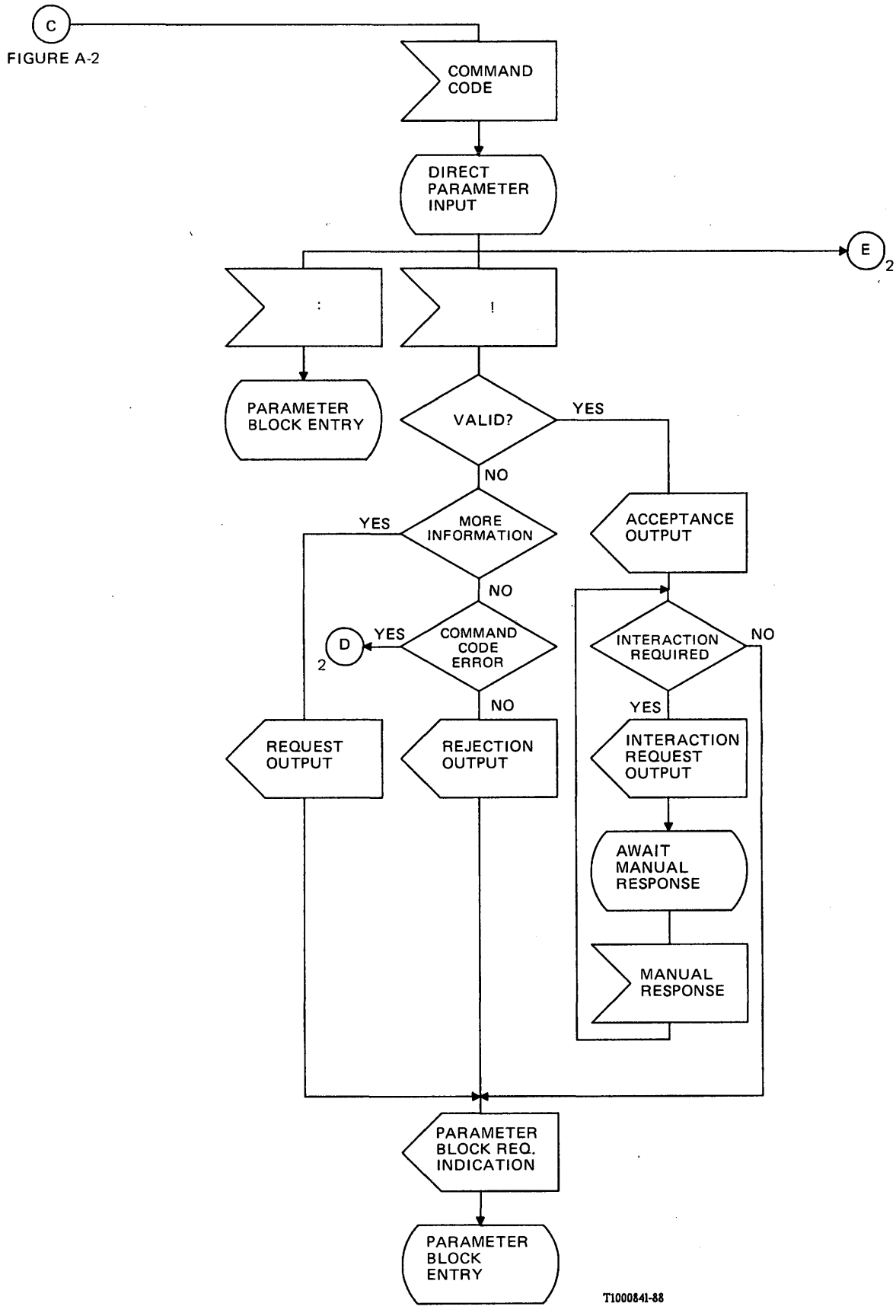


FIGURE A-3/Z.317 (1 of 2)

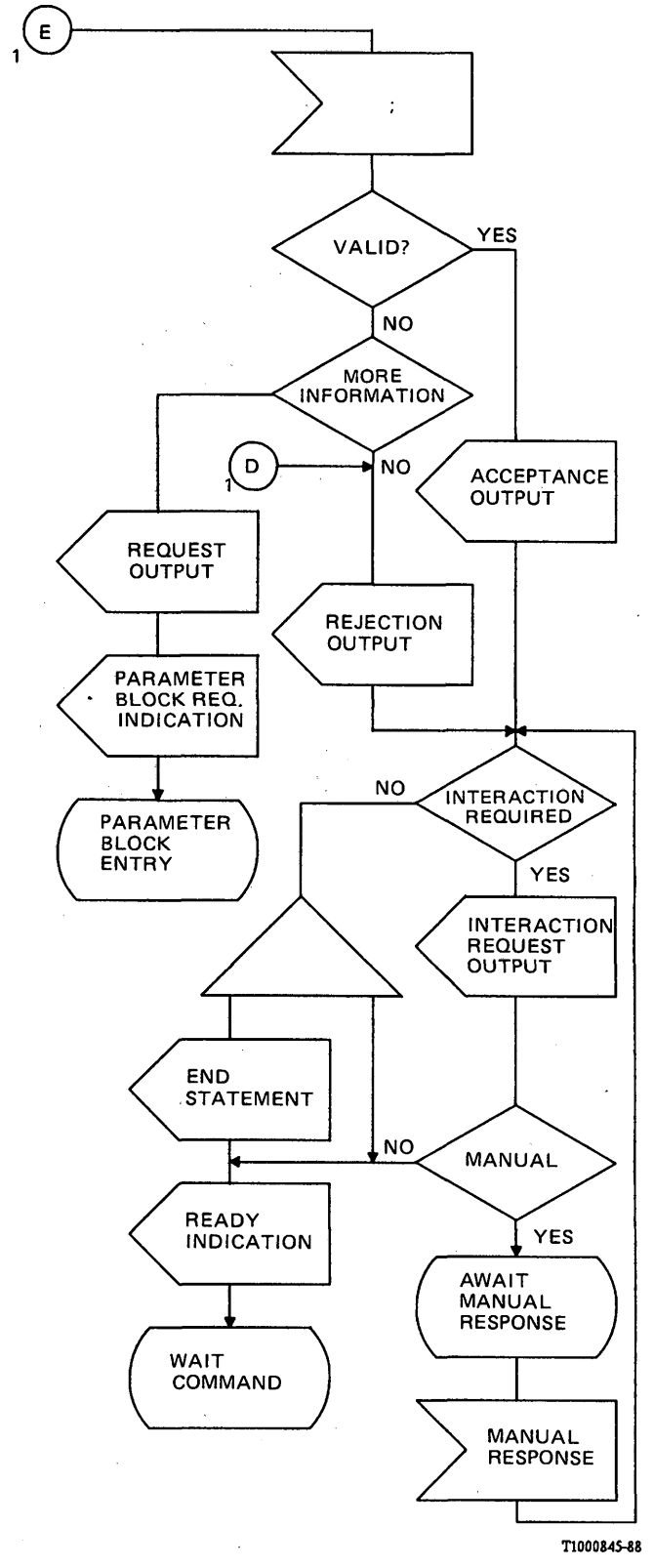


FIGURE A-3/Z.317 (2 of 2)

SECTION 3

EXTENDED MML FOR VISUAL DISPLAY TERMINALS

Recommendation Z.321

INTRODUCTION TO THE EXTENDED MML FOR VISUAL DISPLAY TERMINALS

1 Scope of the Section

This Section deals with man-machine interfaces that take advantage of the input and output facilities usually available on visual display terminals (VDTs). The procedures described are not necessarily confined to this type of terminal; they can also be applied to printer-oriented terminals, such as teletypewriters, within the limits imposed by the facilities available at those terminals, e.g., information entry through menu selection.

By maintaining consistency with Recommendations Z.311-Z.317, these Recommendations facilitate a transition from a man-machine interface using basic syntax and dialogue procedures as described in Section 1 to one based on VDTs.

Diagrams and examples are used to clarify and illustrate the concepts explained in the text. The diagrams do not include exceptional cases and do not specify all possibilities available with the extended MML; those not shown diagrammatically, but which are allowed in the text, are subjects for further study and are not excluded from the extended MML. Similarly, the examples shown are not intended to imply a particular system implementation.

The Recommendations cover aspects of VDTs that users see and use, e.g., data entry, data display, interactive control, user guidance, etc. Specific terminal characteristics are avoided wherever possible.

2 Organization of Section 3

Section 3 consists of the following Recommendations:

Z.321 Introduction to the extended MML for visual display terminals

Z.322 Capabilities of visual display terminals

Z.323 Man-machine interaction

Recommendation Z.322 describes many of the capabilities currently available in VDTs. *Recommendation Z.323* focusses on actual man-machine interactions (i.e., *how* the capabilities are used) by addressing various aspects such as dialogue elements, monologue outputs, user assistance and interactive control.

3 Human factors

3.1 *The human factor view of the man-machine interface*

Human factor science characterizes the man-machine interface as any part of a system that the user comes in contact with – either physically, perceptually or conceptually. The user's conceptual model of a system is the knowledge that organizes how the system works and how it can be used to accomplish tasks. The conceptual model forms an integral part of the user interface.

3.2 *The need for human factors considerations*

The aim of human factors is to satisfy the largest possible proportion of potential users rather than to tailor the system to one user, particularly one with a detailed and sophisticated knowledge of the system. Therefore a proper man-machine interface takes account of the user's needs as well as system requirements. Poor quality will show up as a high proportion of input errors, loss of user confidence and motivation and high training costs. A high quality man-machine interface is based on a truly representative user model.

Recognized human factors literature has been used in the formulation of Recommendations Z.322 and Z.323. Where appropriate, human factor aspects have been incorporated into the texts.

Recommendation Z.322

CAPABILITIES OF VISUAL DISPLAY TERMINALS

1 Introduction

This Recommendation describes some of the capabilities which are important to the user and which are commonly available on interfaces based on VDTs. It is not an exhaustive list of capabilities. The use of additional capabilities, not covered in these Recommendations, is not precluded. Not all of the capabilities described need be present on a given system. Graphics capabilities are for future study and are therefore not considered in detail in these Recommendations.

System implementation of these capabilities may vary, depending for example on the degree of intelligence in the terminal itself and the allocation of responsibility for the man-machine interface among system components.

Items covered are treated from the point of view of the importance of their characteristics for designing the man-machine interface. Human factors are dealt with individually for each item.

2 Screen

2.1 *Character definition*

For further study.

2.2 *Cursor*

The cursor is important in the operation of a VDT because it directs the user's attention to that position on the screen appropriate to the task at hand, e.g., where the next character will appear. The cursor also allows the user to specify the location on the screen where an entry or change is desired by the user.

A set of general cursor qualities include:

- easily found by the user at any position in the display;
- easily tracked as it is moved through the display;
- does not interfere with the reading of the symbol that it marks;
- should not be so distracting as to impair the search for unrelated information displayed elsewhere on the screen;
- should be of a form that is unique and reserved for that purpose only;
- should be stable in respect to the position to which it is addressed until it is readdressed elsewhere as a result of user or system action.

2.3 *Screen partitioning definition*

The following definitions describe the physical partitioning of the VDT screen.

2.3.1 *Visible display*

The visible display is the entire physical screen of a VDT (see Figure 1/Z.322).

2.3.2 *Border area*

The border area is that part of a visible display which is physically unavailable for displaying or entering data (see Figure 1/Z.322).

2.3.3 *Display area*

The display area is that part of a visible display which is available for displaying or entering data (see Figure 1/Z.322).

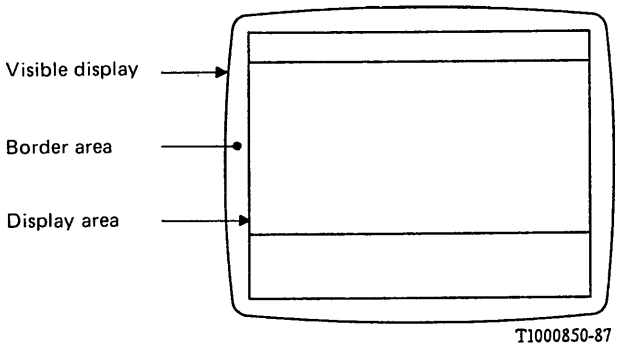


FIGURE 1/Z.322

Screen partitioning

2.3.4 *Window and window area*

The display area can contain one or more windows. A window contains a collection of related information. A window can consist of one window area or can be partitioned into window areas.

The different characteristics and operations specifying windows and window areas depend both on the system type and on the physical capabilities of the terminal.

2.3.4.1 *Window definition*

A window is a collection of one or more window areas which can occupy a part of the display area (sometimes the entire display area) and it is used for entering and/or displaying data. The collection depends on the application. A window is dedicated to an application. More than one window per application can be present at the samne time on the display area.

2.3.4.2 *Window characteristics*

The main characteristics of a window are:

- its name: allowing it to be identified;
- its location: relation to the other windows in the display area. Windows are displayed independently of each other. Windows can appear superimposed – one on top of another – or located side by side. When a window is located on top, it can hide a window or windows that are below it;
- the list of window areas it can contain;
- its size: its size expressed as height and width can vary;
- its state: a window can be "interactive," or "not interactive". Information entry can be performed only when the window is "interactive";
- its visibility: a window is visible when it appears totally or partially on the screen. It can be partially visible either because it is overlapped by another window or because a part of the window is outside the display area;
- its limits: when it is visible, the limits of a visible part of a window must be obvious to the user;
- the application it is dedicated to.

2.3.4.3 *Window area definition*

A window area is a named part of a window that is dedicated for a specific purpose depending upon the application.

2.3.4.4 *Window area characteristics*

The main characteristics of a window area are:

- its name: allowing it to be identified;
- the purpose related to it;
- its presence state: a window area can be "present", or "not present". If a window area is "not present", it can not be seen on the screen whatever the position of the window it belongs to;
- its position in the window: the relative location of the window areas in a window should be fixed. This location can only be modified by changing the presence state of other window area(s);
- its size: its size expressed as height and width may vary;
- its visibility: when a window area is present, it can appear or not appear on the screen depending whether the part of the window it belongs to is visible or not;
- its limits: when it is visible, the limits of a window area must be obvious to the user;
- its text management: scrolling can be available in a window area.

2.3.4.5 *General rules for the display of windows and window areas*

A window can appear anywhere on the screen totally or partially in a non-restrictive manner.

Windows and window areas need not be displayed in all systems or in all applications or all the time in a given system.

The limits of windows and window areas must be unambiguously clear to the user. The techniques that may be used to achieve this include, but are not limited to the following:

- lines and boxes;
- inverse video;
- background colour. This use of colour should be distinguished from the use of colour as a highlighting technique where colour should be used in combination with other techniques.

Some examples of screens illustrating the use of windows and window areas are given in Figures 2/Z.322 to 5/Z.322. In these figures, windows are outlined by double-line boundaries while the boundaries between window areas are depicted by single lines. Lines and boxes are used simply as a concrete example that can be produced easily in print.

2.3.5 *Field*

2.3.5.1 *Field definition*

A field is a part of a window (sometimes the entire window area), which is used for entering or displaying information.

2.3.5.2 *Field characteristics*

The most important characteristics, which may vary in time, are:

a) its position within the window area;

b) its size;

c) its type:
   - for entering information (input field): accessible for writing by user and system (e.g., default value);
   - for displaying information (output field): inaccessible for writing by user.

The limits of an input field must be obvious to the user. There may be one or several fields within one window area (see Figure 6/Z.322).

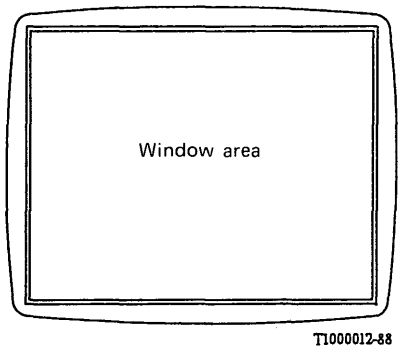


FIGURE 2/Z.322

1 window containing 1 window area

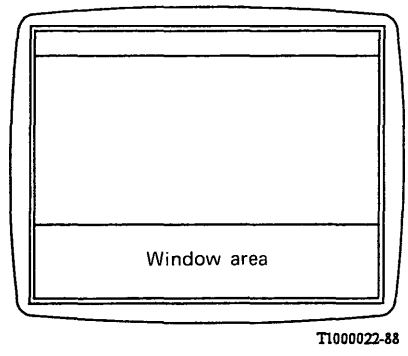


FIGURE 3/Z.322

1 window containing 3 window areas

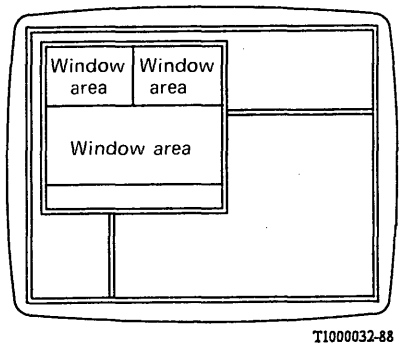


FIGURE 4/Z.322

2 overlapping windows one of which is partially displayed in the display area

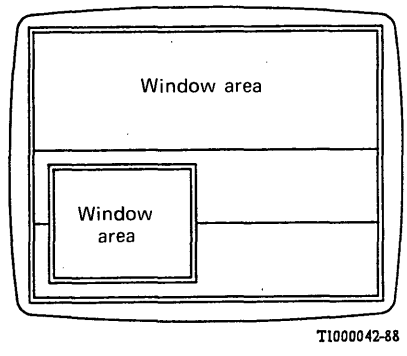


FIGURE 5/Z.322

2 windows

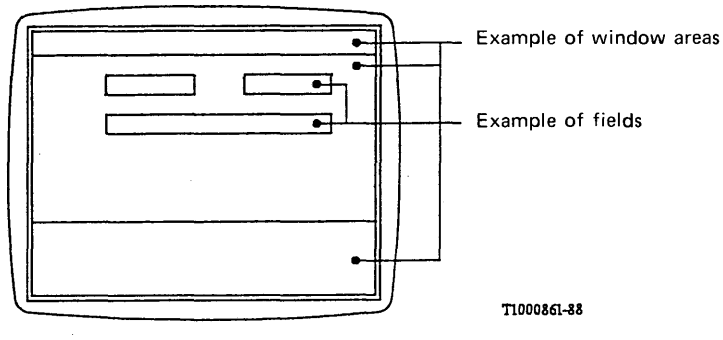


FIGURE 6/Z.322

Fields in a window area

2.4 *Physical characteristics*

For further study.

2.5 *Video attributes*

Video attributes are used to emphasize certain important information, e.g., a title, a message, a chosen item, in order to attract the attention of the user. Video attributes work on the characters of the information shown within an entire window or window area, a part of a window or window area, an entire field or within just a part of the field.

The following video attributes may be provided singularly or in combination:

2.5.1 *Luminance*

For further study.

Information can be displayed in different levels of luminance.

2.5.2 *Colour*

Information can be displayed in different colours.

2.5.3 *Flashing*

Information can be displayed alternately as normal characters and as spaces in the prevailing background colour.

2.5.4 *Underline*

Information can be displayed with underlined characters.

2.5.5 *Size*

Information can be displayed in different character sizes.

2.5.6 *Font*

Information can be displayed in different fonts, e.g., italics, bold.

2.5.7 *Inverse video*

Information can be displayed by inverting the image of the characters, such as going from light characters on a dark background to dark characters on a light background.

2.5.8 *Concealment*

Information can be displayed as space characters, e.g., secret parts of a password.

3 Other output devices

For further study.

4 Keyboard characteristics

For further study.

5 Other input devices

For further study.

6 Transmission characteristics

There are two fundamental transmission mechanisms commonly employed and referred to as "character mode" and "block mode".

If a terminal uses character mode transmission, each and every character input at the keyboard is sent to the controlling processor one at a time. Thus, as is the case with the syntax of Recommendation Z.315, if certain regular keys have special meanings ascribed to them e.g., ; or !, then they can act as specific triggers to the controlling software which then performs some process on the preceding information in accordance with the given syntax rules.

If the same terminal uses block mode transmission, all of the regular typewriter keys and some of the special purpose keys only have an effect local to the terminal, i.e. the information input goes into the "memory" of the terminal and onto the screen normally, but not to the controlling processor. The implication is, obviously, that special actions assigned to these keys do not get processed until an explicit "send" action is made. A "send" action by the user is only required when information is to be moved from the terminal to the host processor.

The important point for the purposes of these Recommendations is that the use of a "send" key is not explicitly shown at any time. It is recommended that systems that employ "block mode" transmission either convey very explicit instruction on when a "send" action is required of the user or are designed to be able to accept and respond intelligently to incomplete input, i.e. "send" can be used by the user at any point without a fundamental disruption of the dialogue. As far as possible this will shield the user from the effects of the transmission mode employed.

7 Control functions

Control functions are those functions related to the man-machine interface that are applied by the user independently while in a dialogue with the system functions. Control functions have no direct impact on the system functions. Control functions are subdivided into cursor control functions and interface control functions.

7.1 *Cursor control functions*

A cursor is generally used as an indicator of the position where an action will take place, such as a character being written on the screen – either by the system or by the user. Cursor control functions do not directly affect the overall system state, but assist users in selecting data entry fields, editing fields, etc.

Examples include:

a) *Home position of the cursor*

Here "home" means a position in the display area to which the cursor can be consistently moved, from any position, by a single keystroke. The actual position in the display area which represents "home" may vary according to the activity being performed and the current layout of the display area.

b) *Movement control of the cursor*

Assuming that the VDT used supports direct cursor addressing, the following types of cursor movement are possible:

i) by the system, and

ii) by the user via cursor control functions. General cursor control functions independent of dialogue are:

- one line up;
- one line down;
- one place left;
- one place right.

Ideally, cursor movement should be easy to accomplish by means of a single, dedicated key for each function. Shifted characters should be avoided. If a cursor positioning control key is used, it should repeat when held down. Cursor movement may also be controlled by other input devices, e.g., light pen, trackball, mouse or joystick.

When cursor positioning is incremental by discrete steps, the step size of cursor movement should be consistent in both right and left directions and both up and down directions. However, the cursor may bypass inaccessible fields.

When character size is variable on the display, incremental cursor positioning should have a variable step size corresponding to the currently selected character size.

7.2 *Interface control functions*

Functions of this class are used to force specific actions relating to the interface. They are invoked by various means, including pressing dedicated control keys.

Examples of man-machine interface control functions include, but are not limited to:

- send (other words for the same function are "transmit" and "enter") [see § 6];
- editing control functions (insert character, insert line, replace character, etc.);
- capitals lock (the condition where letters are input as capitals only);
- select different font [see § 2.5.6];
- select different character size [see § 2.5.5].

Recommendation Z.323

MAN-MACHINE INTERACTION

1 Introduction

This Recommendation describes *how* interactions should take place between the user and the system from a logical viewpoint. It describes how an effective man-machine interface should appear to the user when utilizing the capabilities of VDTs as described in Recommendation Z.322. This Recommendation supersedes Recommendations Z.311-Z.317 for interfaces based on VDTs, referencing parts of them where appropriate. Specific human factor guidelines are included within the appropriate divisions of the text.

The capabilities of VDTs, e.g., multiple windows, inverse video, etc., when used consistently can lead to a more effective man-machine interface. Additional dialogue procedures are possible and often preferable with VDTs, e.g., using different windows for different applications. Likewise, the transient nature of information presented on a screen may affect the selection of information display and the manner of presentation. The terminal capabilities available must be considered in conjunction with guidelines presented in this Recommendation in order to produce the most effective interface.

Many advances in the state of the art of man-machine interface design are incorporated in Recommendation Z.323. However, the use of graphics capabilities have not yet been considered in any detail in these Recommendations and must be studied further. The needs of the user moving between different systems or different types of terminals are best facilitated by ensuring that capabilities are used consistently and that user assistance is an integral part of interface design. Interfaces designed according to the principles outlined in this Recommendation will tend to be more user friendly, effective interfaces.

2 Common aspects

2.1 *Data display*

Data display is the presentation of information by the system to the user. During a dialogue the number, dimension and position of windows, window areas and fields in the display area may be changed. Not all fields, window areas or windows need necessarily display information at any one time.

Visual display terminals facilitate information entry through menu selection and form filling. Since presentation of more information at one time might cause confusion, care must be taken to label information clearly, to keep displays simple, to highlight information consistently and in moderation, and to maintain a consistent information layout as far as possible.

2.1.1 *General guidelines*

The layout of output is dependent on what type of data is presented. There are three basic types, combinations of which are possible:

- textual data;
- numeric data;
- tabular data.

a) *Guidelines for textual data*:

- text should be written using upper and lower case letters;
- abbreviations should not be used if confusion might be caused;
- plain text should be used rather than codes.

b) *Guidelines for numeric data*:

- strings of more than five numeric characters may be presented in groups of two to four;
- standardized formats (e.g., data and time as specified in Recommendation Z.316) should be used.

c) *Guidelines for tabular data*:

- in case of lengthy columns, spacing between about every five items improves readability;
- items which are related to each other should be placed close together;
- figures arranged in columns are easier to compare than figures arranged in a row;
- integers should be right justified;
- numerical entries with decimals should be justified with respect to a fixed decimal point position;
- text and labels should be left justified;
- if any text continues on another line, it should begin in the same column as the text above.

2.1.2 *Accessible and inaccessible parts of the display area*

VDTs provide the capability to characterize some fields of the screen as accessible for writing by the system only, some other fields accessible for the system and the user.

The fields used for the display of headers, parameter identities, delimiters, etc., should be accessible for writing only by the system (output fields). The fields used for the input of parameters should be accessible both to the system and to the user (input fields). The system can highlight these fields, for example, by underlining to distinguish the field or a default value, if appropriate. The user can access the field to input the desired value(s), to edit the previous input value(s), or to edit the offered default value.

The user may attempt to write into a field reserved for the system. This should not be allowed, an indication should be sent to the user and the input characters should be ignored. The type of this indication depends on the terminal facilities and may be an audible or visible signal. However, the terminal shall immediately recover from this situation so that the user can proceed.

2.1.3 *Highlighting*

Highlighting is used to emphasize visually a portion of a display area to make it stand out from adjacent portions, i.e., to call the viewer's attention to it. It should be used consistently and in moderation. In particular, care should be taken not to confuse or otherwise overload the user by highlighting.

There are a number of areas where highlighting may be applied, such as:

- defaults in forms;
- optional information entry in forms;
- indication of system irregularities and their urgency, etc.

There are a number of possible highlighting techniques, such as:

- different levels of luminance;
- colour;
- flashing;
- underlining;
- different character sizes or fonts;
- small or capital letters (lower or upper case);
- pointing with arrows, asterisk, etc.;
- inverse video;
- combinations of the above.

Some guidelines that should be followed in all applications of highlighting are:

a) when colour screens are used:
   - in order to reduce problems for colour-blind users and to facilitate a transition between colour and monochromatic terminals within the same system, colour should normally be used in combination with some other means of distinction. Note also that some colours may have psychological associations, perhaps depending on the cultural tradition of a nation, e.g., red for danger, green for proceed;
   - be consistent in the use of colour. Colour is a means to recognize rapidly particular windows, window areas or fields on the screen, independent of any system;
   - colour should be used for additional distinction and emphasis. For example, colour should be used for aiding the user in locating information and for alerting the user to status changes. Colour should be used sparingly. It should not be used for purely aesthetic and nonfunctional effect as the main aim;
   - if the user is given the capability to modify the colour of any area or object displayed on the screen, the user should be cautioned about changing colour via any assistance mechanism provided to the user. For example, in the case where the user is changing adjacent areas/objects to the same colour, a warning should be given. Where the capability is provided, the user should be allowed to make any modifications desired. Also, it is desirable that secure access to this capability be provided;
   - the number of colours with specific meanings should be limited. Associating meanings with too many colours may confuse the user;
   - colour combinations should be chosen such that there is sufficient contrast in hue and density wherever two colours meet. This is particularly true in the case where a text is displayed over a colour background;
   - colour combinations should be chosen with care, as many combinations can be displeasing to the eye;

b) use only one level of luminance in addition to the normal level when highlighting. Variations in room lighting, specific VDTs and user perceptions make it unlikely that more than two levels will be universally distinguished;

c) when using more than one highlighting technique, do not highlight more than 30% of the display. If everything is highlighted, even differently, then nothing is highlighted;

d) since flashing attracts much attention, its use should be restricted to special applications, e.g., alarms. Once the user acknowledges the perception of the flashing information, the flashing should be stopped;

e) if the user needs to read text from a flashing area, the flashing should be slow in order to make the text readable. An alternative would be flashing pointers, pointing to the text area of importance;

f) in one system, or at least in each job area, highlighting facilities should be consistently applied;

g) information can be displayed with underlined characters. However, this type of video attribute might make it difficult to observe the cursor on terminals where the underline character is used as the cursor.

2.1.4 *Information layout*

A user should always be able to recognize at first sight:

- where parameter input is desired in a form;
- where system response is expected;
- where the system status is displayed;
- where user guidance is expected, if requested;
- where menus are displayed.

Therefore, the information layout, when determined by the system, should follow common rules in such a way that information of certain categories will be displayed in certain portions of the display area.

The information layout should be consistent in any one system. Information, which is not necessary in certain job areas, may be omitted.

2.1.5 *Description of window areas*

The following window areas can be distinguished in a window on the display area:

- *General information window area.* This window area can contain system identification and/or application identification, and optionally, date, time, and other relevant information. This window area is optional;
- *Status window area.* This window area should contain alarm indicators of the system being controlled, trouble reporting information from connected equipment, and message waiting indicators. The information displayed may be restricted to the particular application being controlled. This window area is optional;
- *Work window area.* This window area should be used for information entry through form filling and menu-item selection. The work window area may also be used as a graphic display and screen editor area, and should support scrolling. This window area is required for information entry through form filling and menu-item selection but is optional otherwise;
- *Output and input window areas.* These two window areas should support scrolling and should be user controllable in size. The input window area should be used for direct information entry. Response to the direct information entry as well as output outside dialogue should appear in the output window area. Input acknowledgements may also appear directly following the command in the input window area. The scrolling should occur in two window areas separately, or both window areas may be combined into one window area. These window areas are required for direct information entry but are optional otherwise;
- *Special keys and directives information window area.* This window area should display function key labels and specifics about the use of directives. This window area is optional.

2.1.6 *Ordering of window areas*

The relative locations of the status, work, output, and input window areas should be fixed for any given system.

Screen layout recomendations for window areas that span the entire width of the window are shown in Figure 1/Z.323. In this case, the screen layout will have the window areas ordered as shown with the understanding that each window area remains optional.

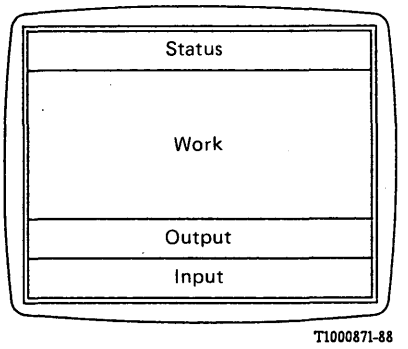


FIGURE 1/Z.323

Window areas that span the entire width of a window

2.2 *Input editing*

Editing mechanisms can be used to correct erroneous input during data entry or to change previously entered input in order to resubmit it.

Several possibilities of editing can be distinguished, including the following:

- delete last character or last n characters;
- delete or overwrite last field;
- delete or overwrite arbitrary fields;
- insert characters.

Editing mechanisms may be dependent on the facilities of a terminal, such as function keys.

2.3 *Response time*

In a system operating normally, response output (see Recommendation Z.317) to a command should be presented to the user within a psychologically acceptable time limit, normally taken to be of the order of two seconds after input. For any given type of command, this time limit should be as uniform as possible in order to meet the expectations of the user.

Depending on the nature of the command, two types of response output can be distinguished:

a) that which conveys the results of the execution of the command;

b) that which concerns the acceptance only of the command, results being communicated to the user by output outside dialogue.

Response output concerning user errors should be given to the user as soon as possible. Although a fixed rule cannot be defined, the following guidelines can be given:

- syntactical errors must be discovered very early by the system; the response time should be within the psychologically acceptable time limit;
- semantic errors can sometimes be discovered early, sometimes late, depending on the type of command and on the nature of the error; normally the feedback should be given to the user as soon as the error is detected;
- semantic errors in pre-scheduled jobs should be indicated to the user either immediately after the command input, if this is possible, or at the time the result is expected.

2.4 *Directives*

The presentation of system output in the form of guidance output, menus, form output, waiting system reports, next page, etc., can be controlled by means of input statements called directives. It is possible to qualify the effect of directives either by the use of context or by the use of additional parameters.

Directives are used to direct the system to present information rather than to execute a command; they can also be used in the interaction between the user and the system prior to command execution.

Directives can be given to the system by a word, e.g., HELP, by a special character, e.g., "?" (question mark), a dedicated function key, or by non-keyboard devices.

Directives can never cause any change in the state of the system. This distinction from commands is made to encourage users to make full use of such facilities without fear of altering the system unintentionally.

The subject of directives needs further study.

2.5 *User guidance*

When a user interacts with a system, there are times when more information about the system is needed than provided by the dialogue element in use, to assist the user in proper and efficient system operation. This information can be provided by means of various categories of user guidance.

Examples of different types of information that could be obtained in a guidance output are:

- how to obtain more specific guidance. A single guidance output at the highest level of simplicity might be displayed when the user enters a directive without any parameters, and the precise nature of guidance required is unclear from the context;
- general principles of dialogue procedure;
- what telecommunications services are available;
- what jobs can be performed;
- a description of structure and application of either classes of commands or a single command in detail. The user must specifically request that such output be displayed, either from the highest level of guidance output or via the parameter on the guidance directive;
- how the job is performed without actually executing it;
- what the user has done so far;
- what kind of entry the system expects from the user, e.g., possible commands, range of a parameter value, example of a correct parameter entry;
- the meaning and consequence of forms, commands, menu items, etc., which are displayed on the screen;
- the syntax or a short explanation of a specific command or job;
- a short description of a specific parameter, e.g., its default value or the permitted range of values.

In order to make the guidance as effective as possible, the following guidelines can be given:

- any guidance provided must be kept up-to-date and accurate;
- guidance should be available in a consistent manner throughout the system;
- unnecessary codes and abbreviations in guidance messages should be avoided.

A classification for user guidance based on user interface characteristics is presented in Figure 2/Z.323.

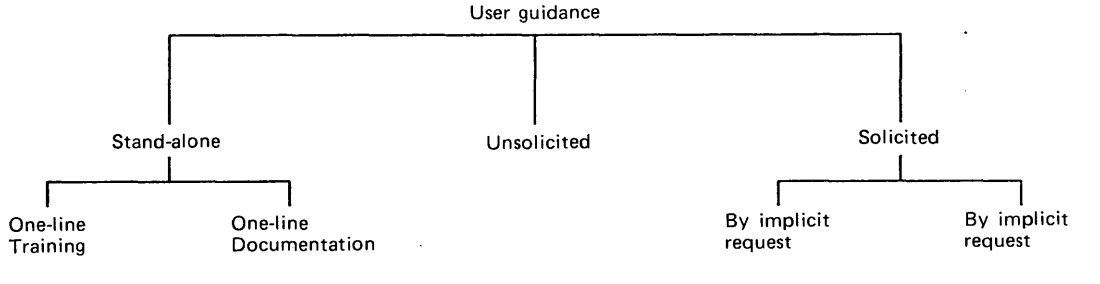


FIGURE 2/Z.323

User guidance

2.5.1 *Stand-alone guidance*

A stand-alone guidance facility can be used without necessarily accessing the function for which the guidance is provided.

2.5.1.1 *On-line training*

The primary purpose of on-line training is to supplement or replace other training methods such as classroom instruction, training manuals, or video courses. It can provide training on how to use the system (or parts of the system) for the first time, to refresh understanding, or to learn the system or function in more depth.

This type of information is provided as a separate function, and is designed to facilitate the learning or educational process.

The major difference between on-line training and other types of guidance is that training usually takes place in a "special" situation, intended to encourage learning. Because of the close relationship between on-line training and other guidance facilities, it is impossible to design or evaluate other guidance facilities without considering the training system.

Rudimentary guidance may be perfect for a trained user who occasionally needs a memory aid, while very elaborate on-line help may be needed for anyone with no previous training.

2.5.1.2 *On-line documentation*

The primary purpose of on-line documentation is to provide the user with a comprehensive body of information about a given subject related to the function. The major difference between on-line documentation and on-line training is that on-line documentation is meant to be used as a reference by users with a fundamental understanding of the function, hence is not a replacement for training. Although available as a stand-alone facility, on-line documentation may also be accessible during execution of the function. In this case, to avoid confusion with other types of guidance, the user should be notified, either implicitly by distinct format or explicitly by a message, that this help is also available as a stand-alone on-line documentation facility.

2.5.2 *Unsolicited guidance*

An unsolicited guidance facility provides user guidance when the system determines there is a necessity. Examples of unsolicited guidance are messages and prompts. Messages are issued to provide information on the current task, give status or completion messages for background tasks, or to notify the user of error situations. Prompts are issued as a result of an action request by the user. Messages and prompts are means through which the system provides feedback to the user, and assists the user in completing a dialogue with the system. They may request specific input, such as a request to the user to key required data, or to request the user to take a specific action, such as inserting a diskette.

2.5.3 *Solicited guidance (on-line help)*

Solicited guidance (also called on-line help) is a system's capability to provide a user with information on how to use the system while using it.

This help facility requires the users to solicit the presentation of help information by means of an explicitly or implicitly stated request. The primary purpose of the on-line help facility is to provide a consistent and easy-to-use tool, that upon request, will give operational assistance necessary so that the user can make efficient use of the system to accomplish a work product.

Help text written using a consistent style is easier to understand and promotes user confidence. The following guidelines are given:

- sentences should be complete and concisely written. Detail should be limited to only what is needed for guidance on the requested item;
- sentences should be action oriented;
- help messages should use familiar wording so that users do not have to learn new wording for familiar concepts;
- references to outside material may be included in the help text, especially if the help information cannot be provided in a concise way.

2.5.3.1 *On-line help by implicit request*

This type of help facility assumes that the user, upon a specific interaction, requests information from the system. The basic distinction between unsolicited guidance and on-line help by implicit request is that the latter can be turned on or off by the user.

For example, the user employs information entry through form filling. If the on-line help by implicit request is activated, the movement of the cursor to a field reserved for the entry of a parameter value causes a message to appear in an output field reserved for implicitly requested help on form filling. The message describes the form in which the parameter value should be entered and the acceptable values. This approach has the advantage that the form layout does not need to be cluttered with supplementary information (as described in Recommendation Z.323, § 3.4.1).

In order to make this type of help facility effective, the following guidelines can be given:

- implicit requests should be limited to accompanying user actions that immediately proceed or are directly related to the entry of information (e.g., moving the cursor to an input field);
- the help displayed as a result of an implicit request should contain concise information that is of immediate use to the user;
- the help message needs to appear in a consistent location which is easily consulted but does not interfere with the information currently in progress;
- the implicitly requested help message should disappear automatically when the user moves on in the dialogue and the message is no longer relevant.

2.5.3.2 *On-line help by explicit request*

This type of on-line help (which will be referred to as "help" in this section for brevity) facility assists the user to complete a work activity by providing specific directions when explicitly requested by the user. The user indicates the item in question, and the system responds with the information specific to the request. Help output is displayed at the user's request by the use of directives.

For systems providing this capability, the following guidelines can be given:

a) *Guidelines on information content and consistency*
   - The information in on-line help should be designed to provide opertional assistance rather than covering training materials, or providing a tutorial;
   - the help should be available within the context of the current dialogue. Contextual help means that within the appropriate authority level, the user can have assistance for items such as menus, options, parameters, commands, objects, or actions relative to the currently displayed information within current task of operation;
   - the type and level of detail of help information provided should be consistent with the anticipated needs of the user at any particular stage of a dialogue. For instance, a "help" request made prior to inputting anything at a terminal could result in a high level introduction to the human-machine interface facilities, whereas a "help" request made instead of inputting a parameter value could result in detailed information on what possible values that parameter could have, and perhaps what each value means;
   - the help facility should be designed to assist the user in progressing from one step within the dialogue to the next by supplying information that gives specific directions the user should follow;
   - the help facility should be available throughout the conduct of any dialogue. For example, if help is available for one menu, appropriate help should be available for all menus;
   - if the user requests help for an item that is not defined within the help facility, the user should be notified that no help is available for the specific item requested and be directed to help relevant to the context;
   - if the system cannot determine exactly what help information is requested, it will present safe information such as a menu of topics instead of guessing at what the user wants;
   - the help facility should allow a user to obtain information about dialogue elements which do not belong to the current context;
   - the help facility should itself have help available. For example, this "help for help" could allow the user to select additional help topics, present a list of possible help items, or provide a brief description of the help facility.

b) *Guidelines on user-help facility interaction*

To provide a simple and efficient interface with the help facility, the following guidelines are given:

- help messages should preferably not overwrite data, error information or user commands and vice versa. In cases where this is unavoidable, a simple mechanism should be provided to retrieve the original information;
- the user interface to the help facility should be consistent with the interface to the other tasks within the system. For example, help menus should be constructed like other menus, selection techniques should operate the same, presentation style should be consistent, and command procedures should function the same;
- when a hierarchy of help information is required, the paths through the hierarchy should be short and simple;
- it should be possible for a user to request directly the exact level of detail required without having to step through intermediate higher level information;
- when possible, the help information should be displayed so that it preserves the visual reference to the dialogue content. Help information is most useful and least disruptive when the user has visual reference to both at the same time;
- where multiple pages of help are available, it should be possible to have any page displayed without having to display intervening pages;
- in the case of a long help message the user must be provided with some means of scrolling back or forth through the displayed text;
- instructions for exiting the help facility should be available on the system;
- when the user explicitly exits the help, the user dialogue should be restored to its original position before the help was requested;
- the help information should remain displayed either until the user explicitly exits the help facility or until the user executes a dialogue step which eliminates the need for the help information.

2.6 *Defaults*

In some applications the normal and most frequently used input can be predicted by the system. Default values which can be considered critical in the sense that they may create situations dangerous to the system integrity should be avoided.

2.6.1 *Use of default values during data entry*

To make the user's work easier, input of the most frequently used parameter values may be prepared by the system. If this offer does not match the user's desire, the possibility to overwrite the default must be open.

An offered default can be accepted by the user, either by active selection such as pressing a dedicated function key or by passive selection, i.e., without taking specific action.

The overwriting or deletion of defaults can be done by editing mechanisms as described in § 2.2.

2.6.2 *Display of defaults during data entry*

The main reason for using defaults is to simplify the user's information entry to the system.

To achieve this, defaults should be offered by the system and may be highlighted as described in § 2.1.3, so that it is obvious to the user which data entry area he has filled himself and which has been filled by the system. The highlighting technique should be consistent in a system or at least in a certain job area.

2.7 *Input error handling*

2.7.1 *Input error information*

In the event of erroneous input, some form of input error information, normally in the form of request output (see Recommendation Z.317), must be presented to the user.

Ideally, input error information would contain:

- where the error was detected;
- what kind of error was made;
- how to recover from it or at least how to find a way to recover from it.

In some cases it may be difficult to supply the user with all this information.

In many cases the input error information may be self-contained, in other cases reference may be made to other sources of information.

The length and detail of the message should be proportional to the nature of the error; the user should not have to look at a long explanation for a simple error.

Coded messages and intimidating jargon such as "syntax error" should be avoided. Messages should be polite and should not patronize or insult the intelligence of the user.

When an error is detected and error information is displayed, the field containing the error may be highlighted.

2.7.2 *Location of error information*

Error information should always appear in a consistent manner on the screen. This should be common within one system or at least within one job area.

2.7.3 *Multiple errors*

Multiple independent errors in one data entry should, if possible, be reported together at one and the same time.

Incidences of conflicting combinations of parameters or parameter values should be treated by the error information as a single subject.

2.7.4 *Correction of errors*

Following detection of an error situation, the user should be provided with mechanisms to correct the erroneous input. Such mechanisms could include:

- the system placing the cursor on the erroneous field and requesting input;
- the user addressing the field, e.g., by name, number or lightpen, or cursor control keys or joystick to get to the field(s) which needs to be changed.

The erroneous information should remain on the screen until it is corrected.

3 Dialogue procedure

3.1 *General*

In the general description of the dialogue procedure, aspects of error correction and of help request are not included. These topics are treated in the detailed descriptions of the specific dialogue elements. For examples of dialogue procedures, see Annex A.

3.1.1 *Structure*

The dialogue procedure is depicted in Figure 3/Z.323.

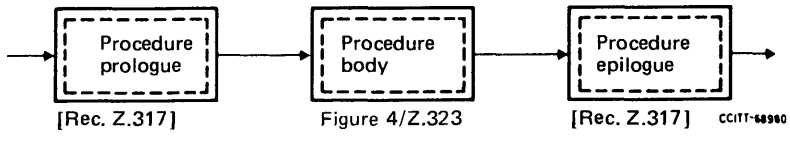


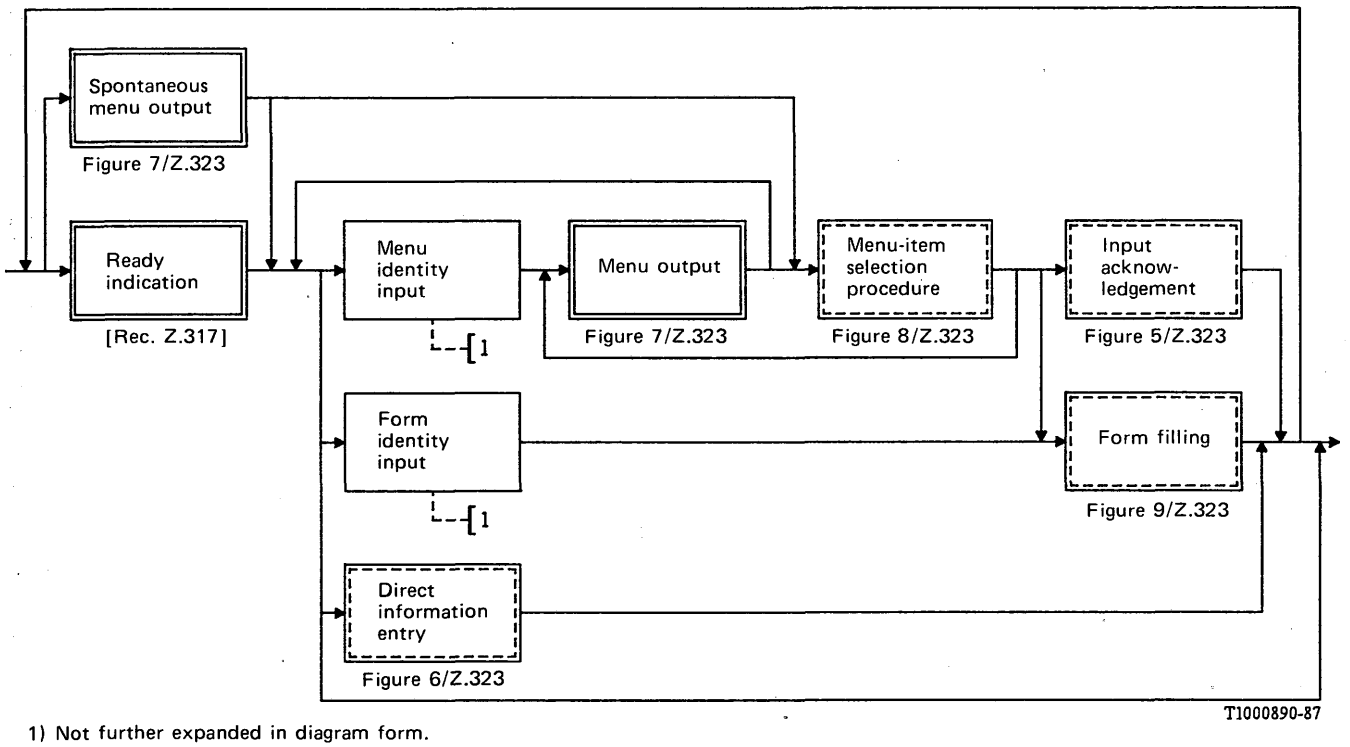
FIGURE 3/Z.323

Dialogue procedure

The dialogue is divided into three main parts:

- prologue;
- body;
- epilogue.

For the procedure prologue and the procedure epilogue, refer to Recommendation Z.317. The procedure body is depicted in Figure 4/Z.323.



1) Not further expanded in diagram form.

FIGURE 4/Z.323

Procedure body

3.1.2 *Dialogue elements*

In the CCITT MML, three different dialogue elements can be distinguished with respect to the method of entering information into the system via a man-machine terminal:

- direct information entry;
- information entry through menu-item selection;
- information entry through form filling.

Information entry can be accomplished exclusively by one of the dialogue elements or – if a system supports more than one dialogue element – by a combination of elements, e.g.:

- menu-item selection and direct information entry;
- menu-item selection and form filling.

3.1.3 *Selection of dialogue elements*

Choosing the right dialogue element depends very much on the nature of the job to be performed and the experience of the user. Often there are many different job areas that the user could deal with during his session at the terminal and the best method, for an inexperienced user, when selecting a job area, and then a specific job in this area, may be to use menu selection(s).

The experienced user would probably prefer a more direct method to reach a specific job, but will also use menu-item selection(s) when performing jobs that are infrequently used. Therefore the availability of both dialogue elements is attractive.

For maintenance staff who gain access to a system via the public switched telephone network with a simple portable terminal, it may not be possible to use every dialogue element due to restrictions imposed by the terminal characteristics.

Directives may be used for selecting dialogue elements. They may be either abbreviated menu or form identities, or function keys. The abbreviated menu or form identities need to be uniquely distinguishable from command codes, e.g., an abbreviated form identity could consist of a command code terminated by a question mark.

If direct information entry is available besides other dialogue elements, then direct information entry should always be possible after output of a ready indication or a menu. This may or may not require the use of a directive.

It should be possible to enter an allowed command or destination identifier even if a displayed menu does not contain it.

3.1.4 *Start and end of information entry*

The system invites the start of information entry by the output of:

- a spontaneous menu (one that is automatically given) and/or
- a ready indication.

The spontaneous menu given may be different depending on the authority of the user or the terminal involved. Any menu can always be requested by the use of a directive.

Completion of information entry always results in an Input Acknowledgement as is shown in Figure 5/Z.323 or in an appropriate error treatment.

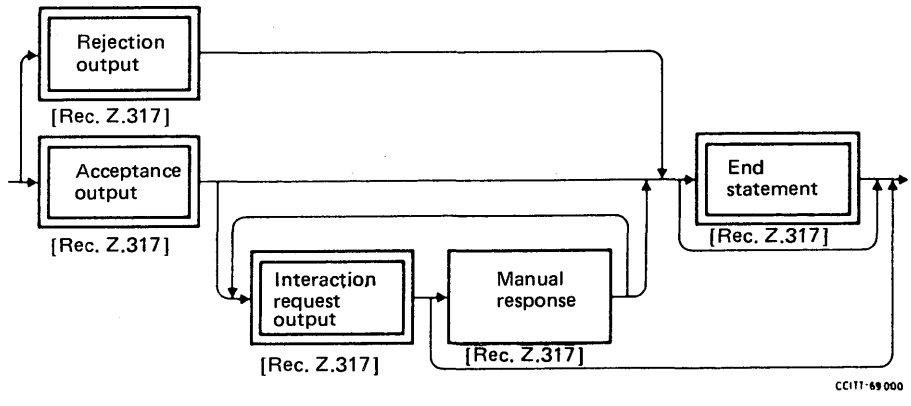


FIGURE 5/Z.323

Input acknowledgement

As in Recommendation Z.317, an Acceptance Output may be followed by an Interaction Request Output.

3.1.5 *End of input indication*

In all dialogue elements the user may need to mark the end of input in order to have the information interpreted by the system. This can be done by some special indicators (see Recommendation Z.314) which contain an implicit end of input indication or by special function keys, e.g., "send". If more than one dialogue element is provided in a system, the end of input indication should be consistently used within each dialogue element.

3.2 *Direct information entry*

Direct information entry can apply to any area of application of the CCITT MML.

The direct information entry, recommended for operation and maintenance, installation and acceptance testing of SPC systems, consists of two sub-elements:

- destination prologue;
- interactive operating sequence.

See Figure 6/Z.323.

For both sub-elements, refer to Recommendation Z.317.

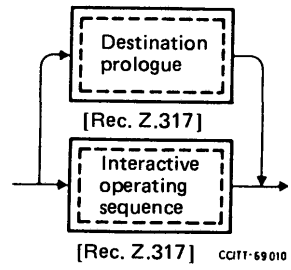


FIGURE 6/Z.323

Direct information entry

3.2.1 *Information entry*

Direct information entry may comprise:

- destination identifier to enable the destination of the information entered subsequently to be changed;
- command code to identify the type of activity to be executed;
- parameter values necessary to allow execution of a requested action;
- manual response as a part of an entering procedure requiring hardware manipulation such as operating switches, replacing equipment, etc.

These aspects are specified in Recommendations Z.315 and Z.317.

3.2.2 *Execution of a command*

A request to execute a command will eventually lead to acceptance or rejection output, refer to Recommendation Z.317.

3.2.3 *User guidance*

Refer to § 2.5.

3.2.4 *Guidance output*

Guidance output is in general related to a command and contains information such as:

- the complete block of parameters to be input for a specific command;
- that part of the block of parameters that is still to be input;
- the parameter next to be input;
- the fact that the complete parameter block has been entered and a request to execute a command can be given.

3.2.5 *Error correction aspects*

Input error information can be contained in guidance output or in request output (refer to Recommendation Z.317 and to § 2.7).

3.3 *Information entry through menu-item selection*

The essential advantage of menu-item selection as a way of interaction is that it can remove memory load from the user. The items available are laid out for inspection, and the way in which each item may be selected is obvious.

The task of performing any transaction using a menu is thus reduced to:

- scanning the items;
- finding the required item (if already known by the user), or deciding which item to choose (if not already known by the user);
- selecting an item.

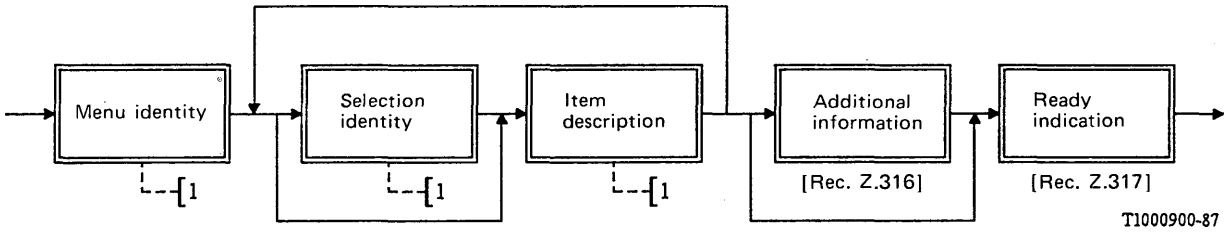
The use of menus is particularly appropriate for applications where there will be many casual users or where there may be frequent interruptions to work at the terminal, and for activities which are infrequently performed.

Menus may be used as a means to arrive at a command code, to select a new destination or to assemble and to execute a command with all its relevant parameters. The system outputs a list of items (the menu output), from which the user can select the appropriate item. In a menu selection procedure, a selection of items from subsequent menu outputs may be needed.

3.3.1 *Display of the menu output*

The menu output (see Figure 7/Z.323) may contain several types of information:

- menu identity;
- menu items;
- additional information.



1) Not further expanded in diagram form.

FIGURE 7/Z.323

Menu output

The information can be displayed in fields and/or given by highlighting techniques.

The *menu identity* is displayed in a field at the head of the menu. It identifies the menu, preferably in a concise meaningful manner to allow an easy recognition of the nature of the menu.

A *menu item* is displayed in a field that contains a brief description of the item and an optional selection identity. By inputting such an identity, a choice can be made. The selection identity should be displayed at the left side of this field.

The *additional information* is intended to present more information to the user in order to aid the selection of an item from the menu, e.g., the text "Enter choice".

The menu layout in the window should be consistent throughout all the menus in a given system. Only one menu should be presented at a time, always displayed in its entirety.

3.3.2 *Item selection*

Refer to Figures 8/Z.323 and 4/Z.323.

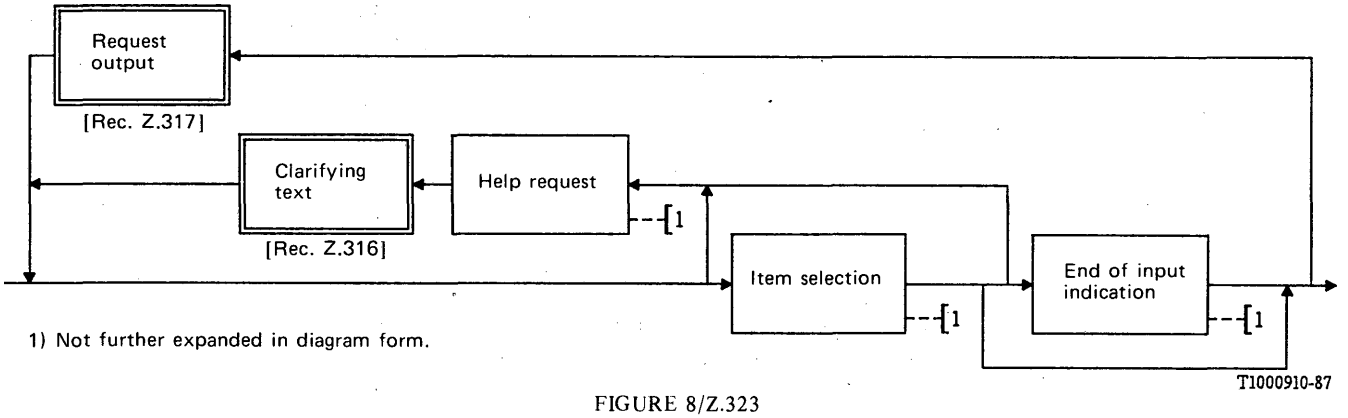


FIGURE 8/Z.323

Menu item selection procedure

The selection of an item can be done in two basic ways:

a) inputting the selection identity;

b) pointing at the item by using techniques such as cursor positioning, lightpen, touch screen, function key, etc.

Selection of more than one item from one menu is not permitted.

When using a hierarchy of menus, it may be helpful for the user to be able to return to the previous menu.

When the user notifies the system that he has made his selection, the system confirms the input by a new menu, a form output, or an input acknowledgement.

3.3.3 *User guidance*

During selection the user can ask for help at any moment. Besides, for general help information the user may ask for specific help information by inputting a specific help request.

The system reacts to the user with a clarifying text (refer to § 2.5).

3.3.4 *Error correction aspects*

The system can ask the user to correct his selection if this is not valid. The response is given in the form of request output (refer to § 2.7).

3.4 *Information entry through form filling*

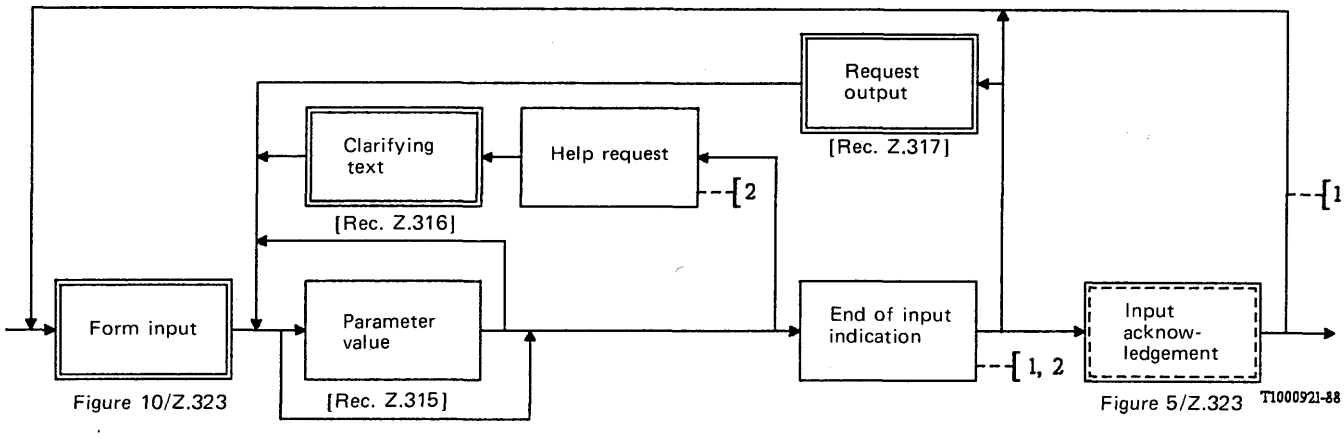
Form filling is a useful method of information entry when flexibility is needed, for example when optional as well as mandatory items of data are required for command execution or handling of data stored in the system.

3.4.1 *Information entry*

When this data entry procedure is to be used, the system first outputs a form (in accordance with Figure 4/Z.323) that requires user input. The form contains a list of parameters identified by parameter identities. The parameter input fields are either empty or contain default values (see Figure 9/Z.323). The form has to be filled in by entering the parameter values required followed by an "end of input indication". For handling data stored in the system, at least the key parameter values have to be input in order to identify the data record. For a read or a delete operation, this is sufficient. For an add or modify operation more parameter values are required. They may partly be obtained by a previous read operation. Completion of the form is indicated by an appropriate "end of input indication".

As many parameter values can be given as desired before an "end of input indication". Parameter value input fields may be skipped if the parameter is not relevant or the initial or existing value is appropriate. Clarifying text is output when a "help request" is input. When the data input by the form is not accepted by the system, a "request output" is given to indicate that completion or correction of the data in the form is required. A sucessful operation is followed by an "input acknowledgement".

The "end of input indication" can also be used to request a next page if the form covers more that one screen. It can also be used to request continuation with a new empty form of the same type after completion. Mechanisms to control this capability are left for further study.



1) Further study is required to address proper provision of user timing control at this point.
2) Not further expanded in diagram form.

FIGURE 9/Z.323

Form filling

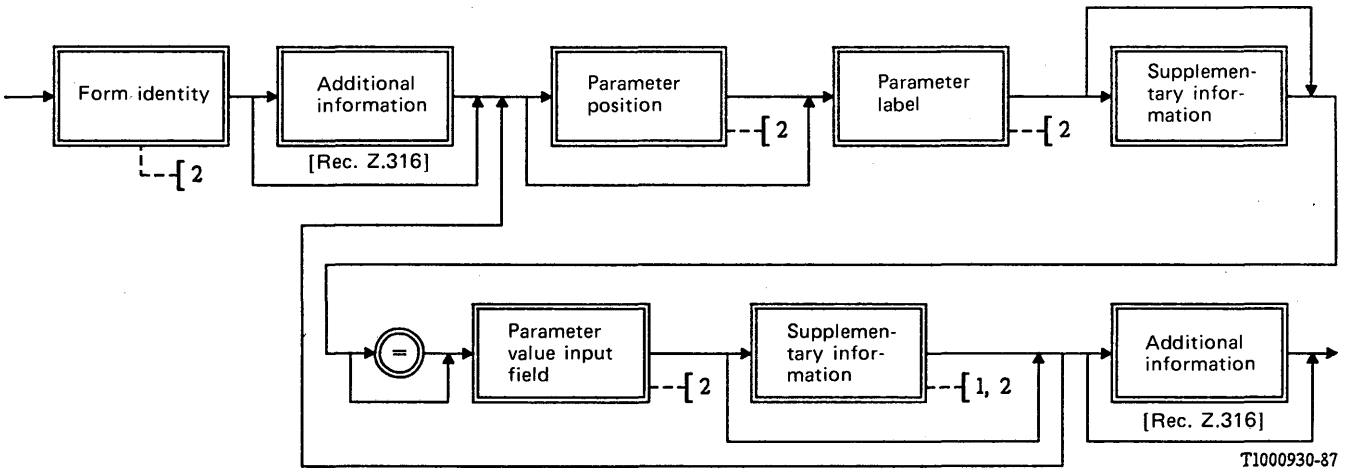
3.4.2 *Form output*

The form output (see Figure 10/Z.323) may contain several types of information:

a) form identity;

b) per parameter:
   - parameter identity,
   - parameter value input field,
   - supplementary information;

c) additional information.



1) Supplementary information may be provided by highlighting the parameter value input field.
2) Not further expanded in diagram form.

FIGURE 10/Z.323

Form output

The above information can be displayed in fields and/or given by highlighting techniques.

The *form identity* is displayed in a field at the head of the form. It identifies the command, preferably in a concise meaningful manner, to allow easy recognition of the nature of the form, and an optional identity for command reference.

The *parameter identity* is displayed in a field and contains the parameter label and an optional parameter position which could be used as a reference in a request output. The parameter label is a text string as defined in Recommendation Z.314. The parameter position should be displayed at the left side of this field.

The *parameter value input field* is an accessible field. Initially this field is either empty and should be filled in by the user, or the system may display in this field the default value which can be overwritten by the user.

The *supplementary information* provides an explanation to the user, if required, to aid input of the parameter value. It may give information such as:

- whether the parameter is optional;
- in which form the value should be entered, e.g., in alphanumeric form.

The *additional information* presents general information to the user with respect to the whole form, e.g., guidance on how to submit the form to the system after finishing the input of parameter values.

The information applying to a particular parameter (parameter identity, parameter value and supplementary information) should clearly be associated with that parameter, i.e., co-located. The position of the fields in the form should be consistent over the form. In any one area of application, they should be consistent from one form to another.

If punctuation is used for delimiting fields, the punctuation from the appropriate direct information entry technique should be used.

3.4.3 *User guidance*

During inputting parameter values the user can ask for help at any moment. Besides general help information he can ask for specific help information by entering a specific help request. (Refer to § 2.5.)

3.4.4 *Error correction aspects*

A consistency check on the set of parameter values in the form should take place after completion. Acceptance or rejection is communicated by "input acknowledgement" or "request output", see Figure 9/Z.323. Validation of value ranges may take place per parameter value input so as to identify range errors as early as possible. Request output as a result of a per parameter check is not shown in Figure 9/Z.323. The cursor and/or highlighting can be used to indicate which value should be corrected. The user can correct the indicated parameter values by changing the values and when complete, re-entering the form contents to the system. (Refer to § 2.7.)

3.5 *Displayed form*

The displayed form can be used to show a form which has already been filled in. The displayed form can only be used for reading and the information cannot be changed by the user. It can appear as an input acknowledgement.

3.6 *Guidelines for the design of menus and forms*

3.6.1 *Scope*

This section deals with the human-machine interface that utilizes the advantage of the input and output facilities offered by menus and forms. By using these guidelines, designers will get a more standardized layout of the various menus and forms.

3.6.2 *General guidelines for menus and forms*

Individual menus and forms should have an identity. Figures 7/Z.323 and 10/Z.323.

Identities should be consistently positioned, preferably on top of the menu or form. (Recommendation Z.323, § 3.3.1 and § 3.4.2.)

Menu and form layout in the window should be consistent throughout all the menus and forms in a given system. (Recommendation Z.323, § 3.3.1 and § 3.4.2.)

Each menu or form should ideally appear in its entirety, so that the user is able to see all the items or parameters at once. If the entire menu or form is not displayed in the window area, then an indication must be given of where the user is in the menu or form.

3.6.3 *Guidelines for menus*

3.6.3.1 *Appearance and organization of menus*

A menu should give hierarchical groupings of logically related items.

A hierarchy of menus should have the least number of levels possible considering the last guideline of § 3.6.2.

Menu items should have a clear and concise description of the choices available. The selection identity should be displayed at the left side of this description.

To avoid errors, special care should be taken to organize and label items in hierarchical menus in such a way that the scope of each item, or the likely result of selecting it, is as clear as possible.

3.6.3.2 *Movement between hierarchical or multiple menus*

If it is possible to go directly to the desired menu by combining menu-item selection identities, then the system should prevent the bypass of mandatory steps.

It should be possible to go backward through the hierarchy, step by step without the necessity of entering the identity of the antecedent menu.

The option to return directly to the main (top) menu should generally be offered.

3.6.4 *Guidelines for forms*

3.6.4.1 *Appearance and organization of forms*

Parameters should be organized into logically related groups. In addition, it may be possible to organize these groups in a hierarchical manner.

Within the primary requirement for good readability, the length of the form should be minimized considering the last guideline of § 3.6.2.

Parameter identities should follow the general guidelines for textual data.

3.6.4.2 *Navigation between input fields in forms*

It should be possible to move the cursor between input fields by a single operation, such as a keystroke. This means that it should be possible to move the cursor to the next or preceeding field in a sequence, or in the case of a form that contains logically related groupings of input fields, it should be possible to jump forward and backwards between the groupings, perhaps skipping several fields.

3.6.4.3 *Presentation of error information about menus and forms*

When errors are made they must be reported to the user in a manner that is most informative, enabling the user to make the quickest recovery.

In some cases it is not advisable to report how to recover from the error, e.g., security reasons.

The location in the window for the error information should be consistent thoughout all the menus and forms in a given system and should clearly be associated with the menu item or the parameter concerned.

4 Monologue output

A monologue output is any output from the system which occurs outside a dialogue. This includes output outside dialogue as described in Recommendation Z.316, system status and alarm information, function key labelling, date and time, etc. Usually, each type of monologue output occurs in an appropriate window on the screen. The occurrence of a monologue output may be accompanied by an audio signal or highlighting in order to stimulate user action, e.g., on alarms. In general, it is not helpful to output information on a VDT which is not immediately useful to the user.

4.1 *Output outside dialogue*

Output outside dialogue is a spontaneous output indicating a certain event, e.g., an alarm situation, or an output in response to a previously entered command, e.g., traffic measurement result. Output outside dialogue should not normally disrupt a dialogue in progress. There are several possible means of achieving this, e.g., message waiting indicators.

4.2 *System information*

System information is information related to the status of the system and may contain items such as:

- system status indicators;
- alarm indicators;
- message waiting indicator.

4.3 *Function key labels*

Function key labels may be displayed in the display area to inform the user as to what functions may be accessed via programmable function keys. They may be displayed as characters or symbols and with various highlighting techniques. It should be obvious to which function key each function key label is associated.

Consistency should be applied when assigning labels to function keys so that frequently occurring labels appear in the same position in the display area.

5 Time-out control inside dialogue

Subsection 5 of Z.317 applies except for the second time-out in which case the timing begins after a spontaneous menu output or ready indication.

ANNEX A

(to Recommendation Z.323)

Examples of dialogue procedures

A.1 *General*

In § 3 of the body of this Recommendation (Dialogue procedure), a number of dialogue elements have been described and Figure 4/Z.323 showing how various inputs and outputs are related has been introduced.

The purpose of this annex is to clarify how the various elements interact. This is done by showing in a number of examples how the interaction between the user and the system appears to the user.

It is important to bear in mind that the examples are intended only to illustrate some of the possibilities described in the dialogue procedure in § 3 of the body of this Recommendation and that they are not to be considered as Recommendations.

In the examples only three types of window areas are shown. These are, from top to bottom: work window area, output window area, and input window area.

The relative position of window areas in the examples is shown in Figure A-1/Z.323. The relative sizes of the window areas in this Figure are not significant, nor are the lines used to delimit the windows. The actual method of best distinguishing windows from each other is terminal dependent.

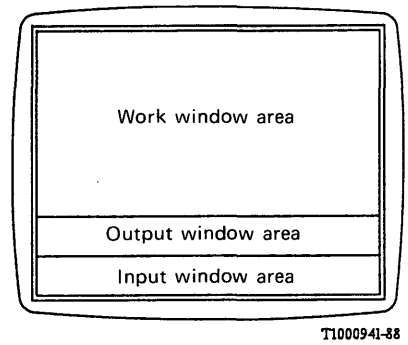


FIGURE A-1/Z.323

Example of window layout

It should be noted that help requests and input error handling are not treated in the examples, i.e., it is presumed that all commands and directives are entered correctly. Each figure shows both the output from the system and the following input made by the user. The user input is written in italics in order to distinguish it from the system output.

Examples 1 though 5 show input of commands, and examples 6 though 8 illustrate data base input.

A.2 *Example 1*

1 The user knows the command code as well as the parameters and enters the entire command by direct information entry.

< COM2: PAR 1=5, PAR 2=10;

2 An acceptance output is displayed and the system is ready for the next input.

Command executed
<

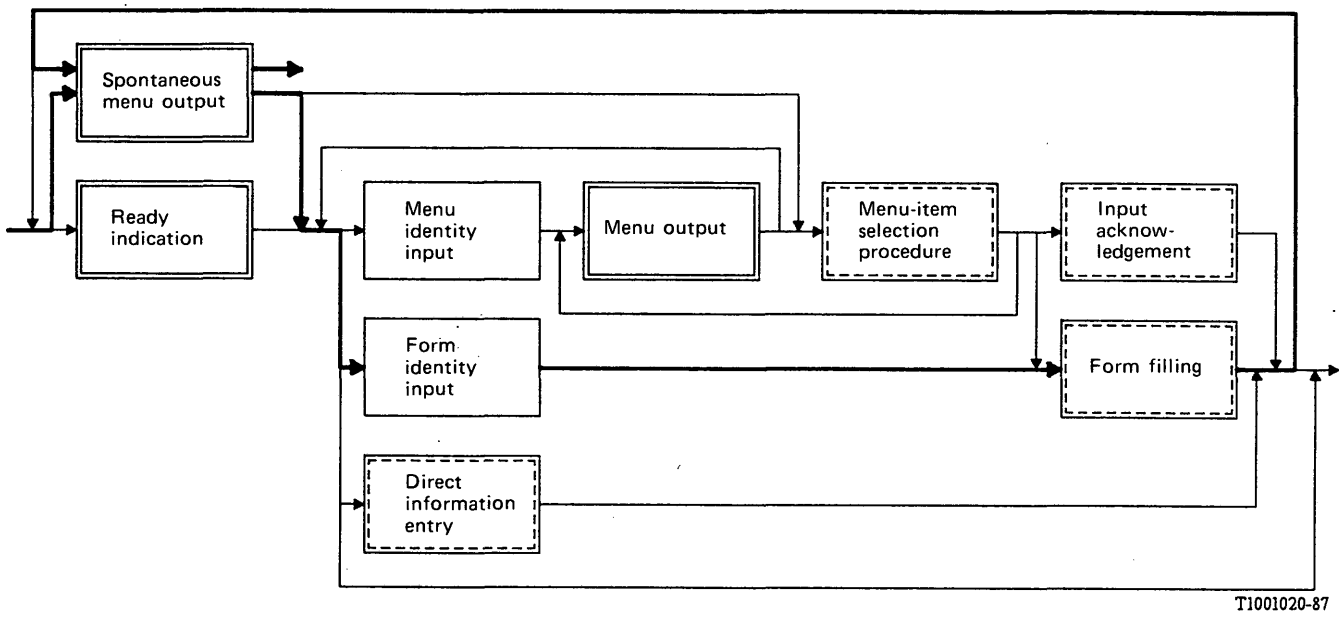


FIGURE A-2/Z.323

A.3 *Example 2*

1 The user knows the command code but not the parameters. He enters a directive in the form of the command code.

< COM 3

2 A form output is displayed. The form is filled and entered. Note that the ready indication is not displayed during form filling. The equal sign is not mandatory.

COM 3 PAR 1 = *560424* PAR 2 = *XYZ* PAR 3 = *100* PAR 4 = *AAAAAA*

3 An acceptance output is displayed in the form of a result and the system is ready for the next input. Note that the output in this example is so spacious that the output window area has increased at the expense of the work window area.

Result – – – – – – – * * * * * – * * * * * – * * * * * – * * * * * – * * * * * – * * * * *
<

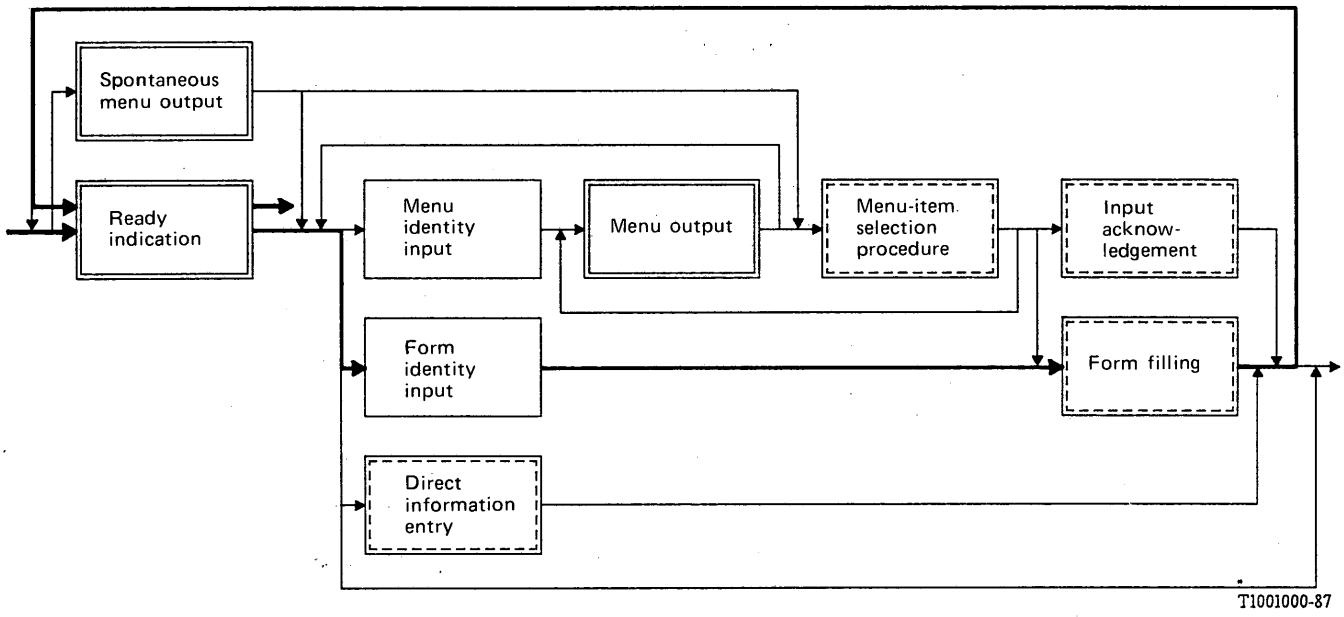


FIGURE A-3/Z.323

A.4 *Example 3*

1 A spontaneous menu output is automatically displayed. The menu items refer to other menus on a lower and more specific level. The user chooses the appropriate menu and enters the associated selection identity.

Menu 1. Menu 1 2. Menu 2 3. Menu 3 4. Menu 4
< *1*

2 A new menu output is displayed in this case the menu items represent command codes. The user selects the wanted command code by entering the associated selection identity.

Menu 1 1. COM 1 2. COM 2 3. COM 3
< *1*

3 A form output is displayed. The form is filled and entered by the user.

COM 1 PAR 1 = *1234* PAR 2 = *GIGA* PAR 3 = *9999* PAR 4 = *500* PAR 5 = *ABCDE*

4 An acceptance output is displayed together with the spontaneous menu. The system is ready for the next input.

Menu 1. Menu 1 2. Menu 2 3. Menu 3 4. Menu 4
Command executed
<

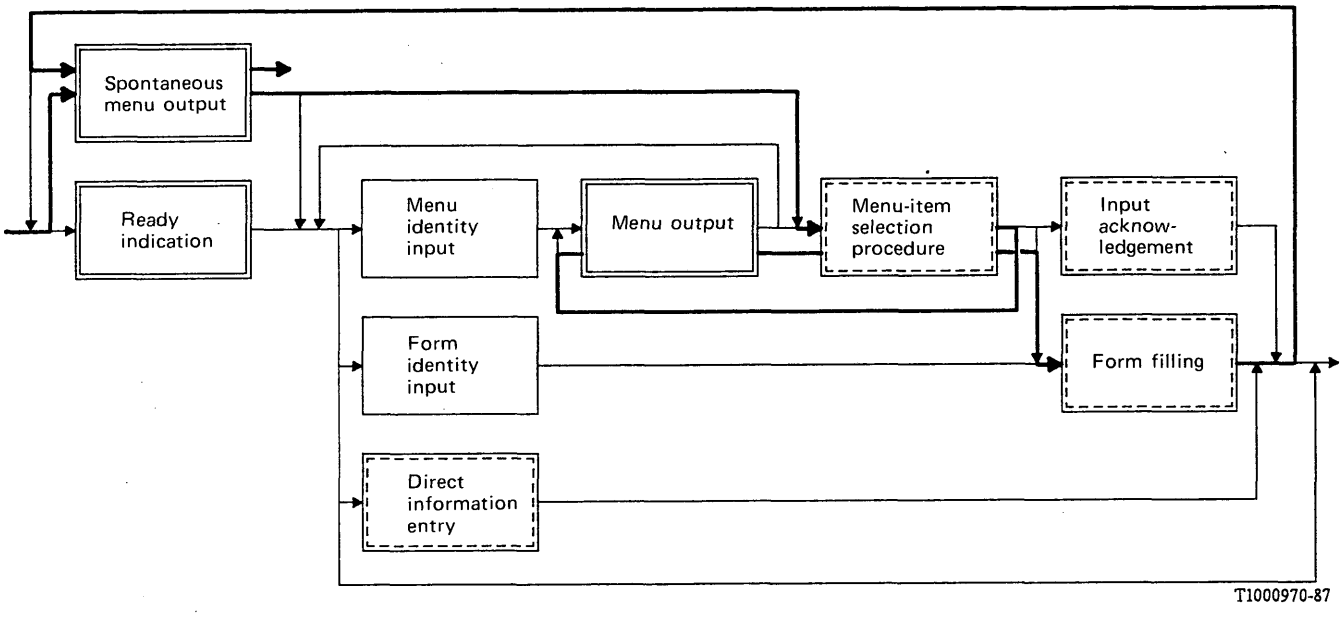


FIGURE A-4/Z.323

A.5 *Example 4*

1 The user enters a directive in the form of a menu identity in order to shortcut to a certain menu.

< MENU 3

2 A menu output containing items referring to other menus is displayed and a selection identity is entered.

Menu 3 1. Menu 31 2. Menu 32 3. Menu 33
< 3

3 The selected menu is displayed. The items in the menu represent command codes. The user recognizes the command code and then remembers the parameters. The entire command is entered directly.

Menu 33 1. COM 1 2. COM 2 3. COM 3 4. COM 4
< COM 2: PAR 1 = 5, PAR 2 = 10;

4 An acceptance output is displayed and the system is ready for the next input.

Command executed
<

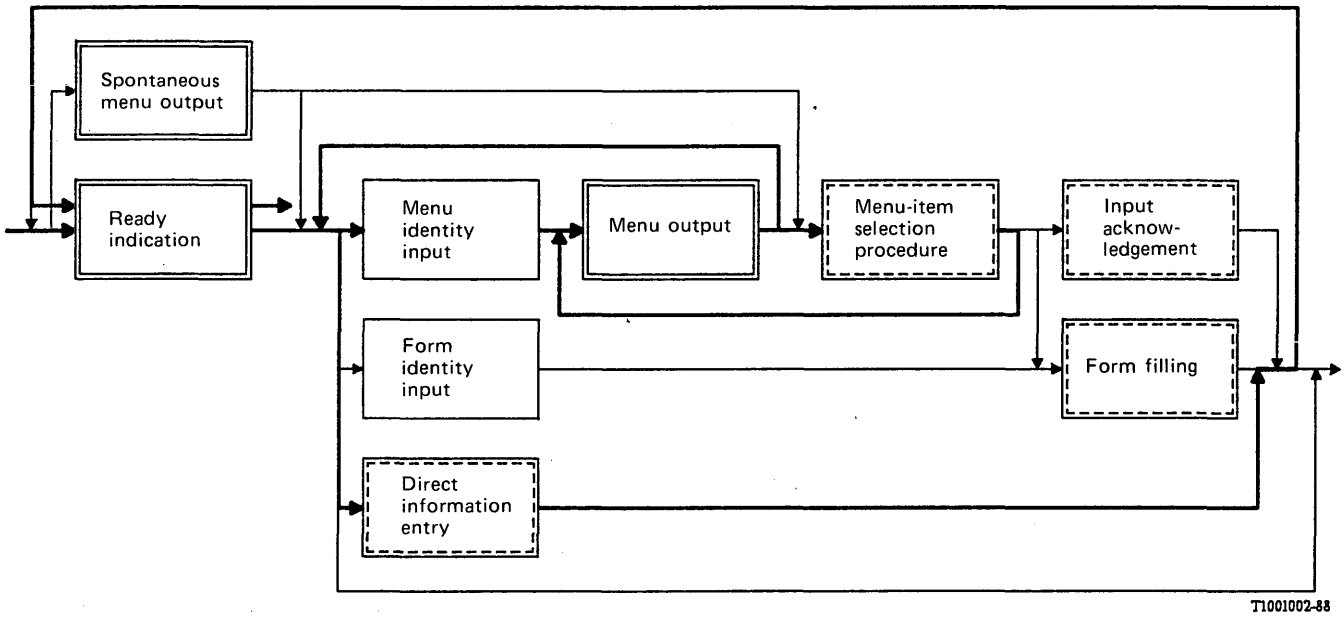


FIGURE A-5/Z.323

A.6 *Example 5*

1 A spontaneous menu output is automatically displayed. The user already knows the command code and enters it.

Note – Cursor positioning is used as a ready indication in this example in place of the "<" character (see Recommendation Z.317, § 3.2.2.1).

Menu 1. Menu 1 2. Menu 2 3. Menu 3 4. Menu 4
COM 4

2 This command requires two forms to be filled in. The first form output is displayed. The user fills in the parameters and enters the form.

COM 4 PAR 1 = *6543* PAR 2 = *GHIJK* PAR 3 = *333* PAR 4 = *XXXXXXX*

3 The second form output is displayed and the user fills in the rest of the parameters and enters the form.

COM 4 PAR 5 = *AEFE* PAR 6 = *LES* PAR 7 = *DIDIT*

4 An acceptance output is displayed. The system is ready for the next input.

Menu 1. Menu 1 2. Menu 2 3. Menu 3 4. Menu 4
Command executed

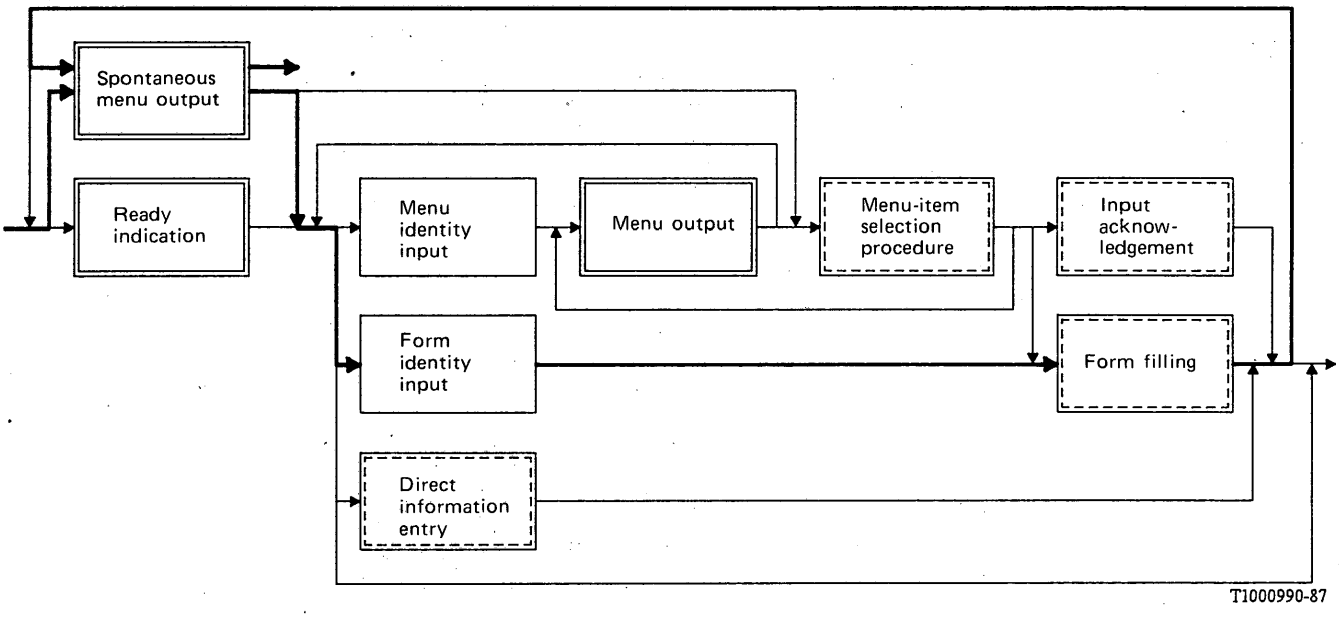
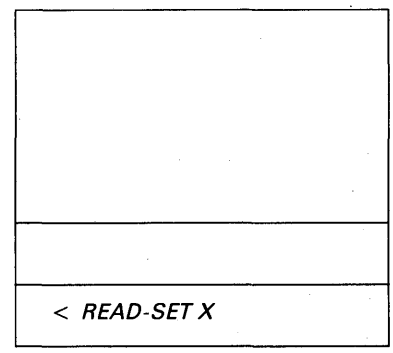


FIGURE A-6/Z.323

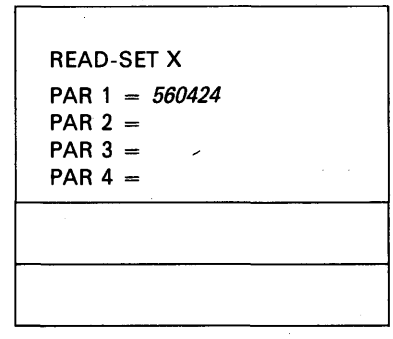
A.7 *Example 6*

1 The user knows the data set and the action to be applied, and enters a directive in the form of a form identity.

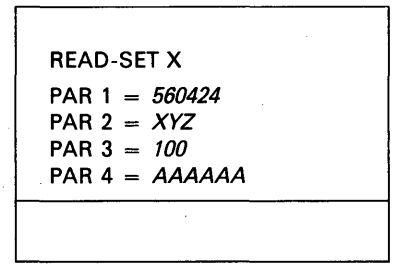
Note – Cursor positioning is used as a ready indication in this example in place of the "<" character (see Recommandation Z.317, § 3.2.2.1).



2 A form output is displayed. The key parameter is filled and entered.



3 An acceptance output is the displayed form in the output window area. The system is ready for the next input.



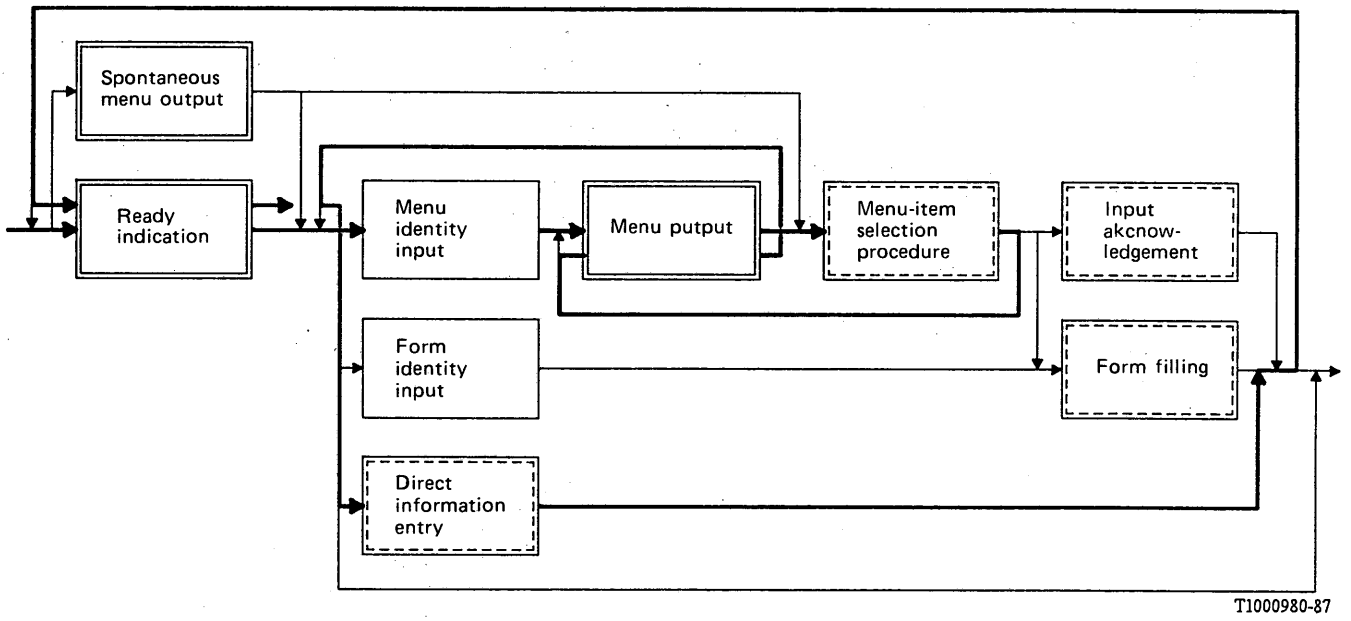


FIGURE A-7/Z.323

A.8 *Example 7*

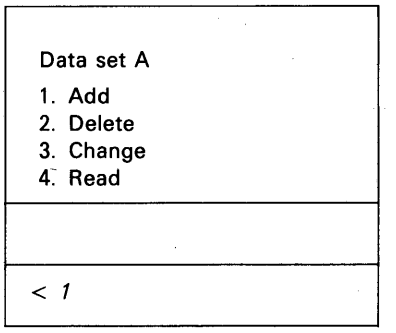
1 A spontaneous menu output is automatically displayed. The menu items refer to other menus on a lower and more specific level. The user chooses the appropriate menu and enters the associated selection identity.

Menu 1. Menu 1 2. Menu 2 3. Menu 3 4. Menu 4
< 3

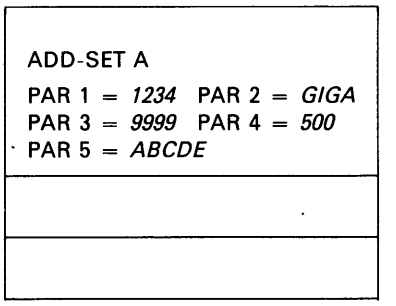
2 A new menu output is displayed. In this case the menu items represent command codes. The user selects the desired action by entering the associated selection identity.

Menu 3 1. Data Set A 2. Data Set B 3. Data Set C 4. Data Set D
< 1

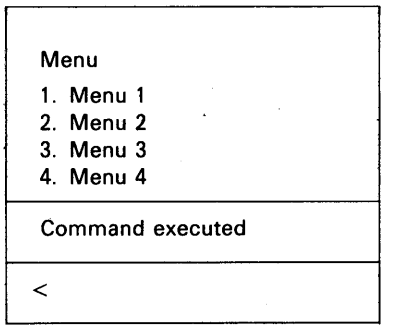
3 A new menu output is displayed. In this case the menu items represent actions. The user selects the wanted action by entering the associated selection identity.



4 A form output is displayed. The form is filled and entered by the user.



5 An acceptance output is displayed together with the spontanteous menu. The system is ready for the next input.



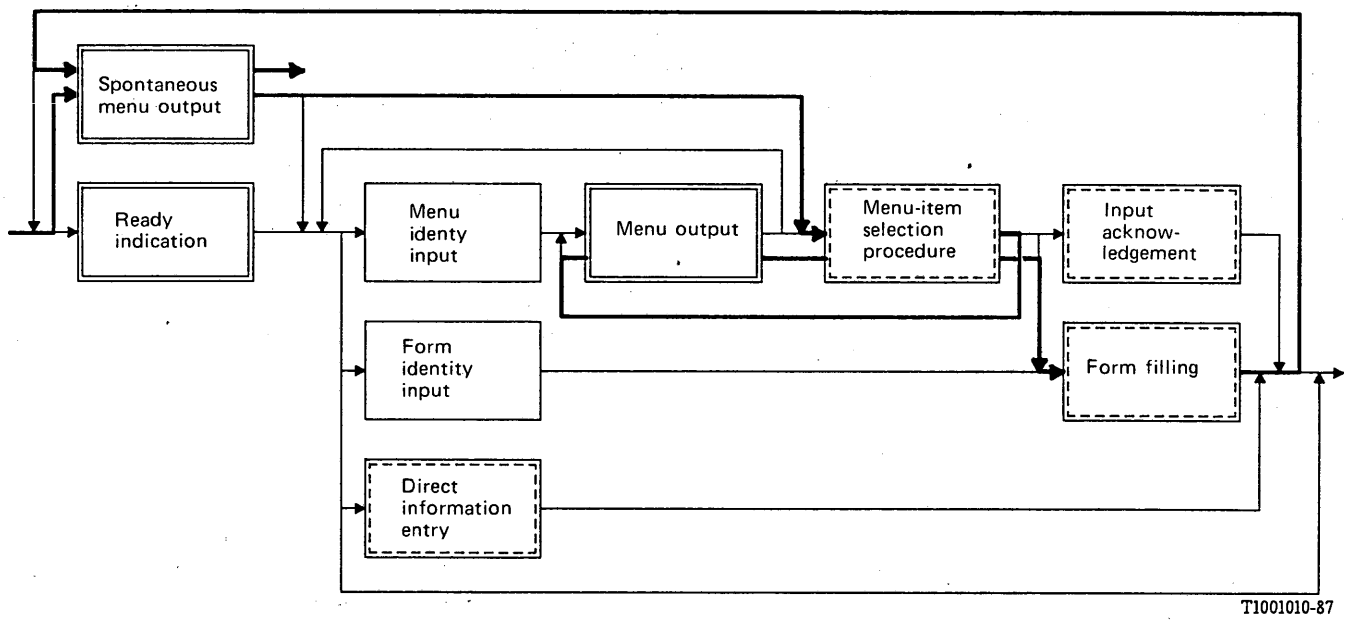


FIGURE A-8/Z.323

A.9 *Example 8*

1 A spontaneous menu output is automatically displayed. The user already knows the combination of data set name and action and enters it.

Menu 1. Menu 1 2. Menu 2 3. Menu 3 4. Menu 4
< *ADD-SET Y*

2 This data set requires two forms to be filled in per record. The first form output is displayed. The user fills in the parameters (data attibutes) and enters the form.

1 of 2 ADD-SET Y PAR 1 = *6543* PAR 2 = *GHIJK* PAR 3 = *333* PAR 4 = *XXXXXXX*

3 The second form ouput is displayed and the user fills in the rest of the parameters and enters the form.

2 of 2
ADD-SET Y PAR 5 = *AEFE* PAR 6 = *LES* PAR 7 = *DIDIT*

4 An acceptance output is displayed. The system is ready for the next input.

Menu 1. Menu 1 2. Menu 2 3. Menu 3 4. Menu 4
Command executed
<

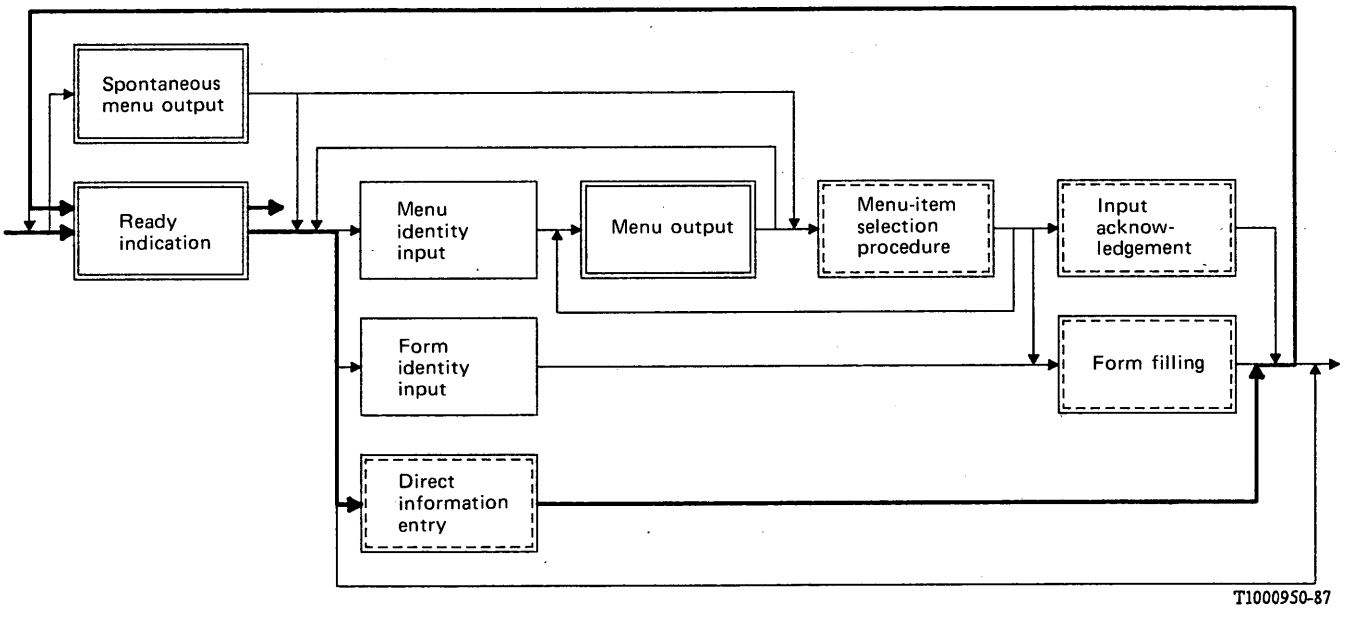


FIGURE A-9/Z.323

ANNEX B

(to Recommendation Z.323)

Examples of windows

B.1 *General*

In § 2.3.4 of the body of this Recommendation a description of windows and window areas are given. (See also Figures 2/Z.323 to 5/Z.323).

The purpose of this annex is to provide some examples of the use of windows and window areas.

It is important to bear in mind that the examples are intended to illustrate the use of windowing only, and that they are not to be considered as Recommendations.

In these examples windows are outlined by double-line boundaries and window areas are outlined with a single-line boundary. This method of depicting windows and window areas is chosen as an example that can be shown easily in print. Actual methods used to distinguish windows will be terminal dependent.

B.2 *Terminal supervision*

This window is related to an application supervising the terminal the user is using. It may contain information about the terminal, about terminal directives (e.g., "window state change" function key), about active connections between the terminal and applications, etc. The window contains two window areas:

- general information;
- output.

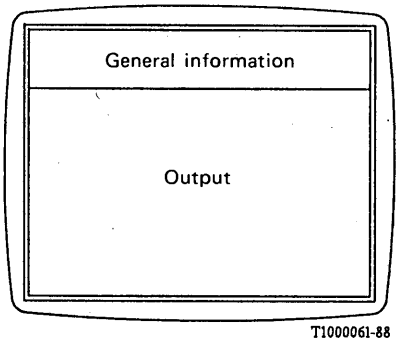


FIGURE B-1/Z.323

Terminal Supervision Window

B.3 *Identification*

This window is related to an application managing the terminals which are local to the site the terminal is linked to. This application performs access connections to terminals with different applications. The window contains three window areas:

- general information;
- work;
- output.

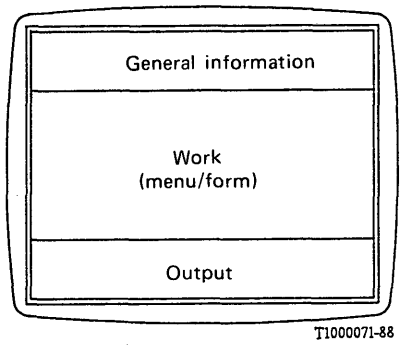


FIGURE B-2/Z.323

Identification Window

In this example, at any given time, the work window area is dedicated to menu/form input.

B.4 *Dialogue*

This window is related to a site operation and maintenance application. It contains four window areas:

- general information;
- work;
- input;
- output.

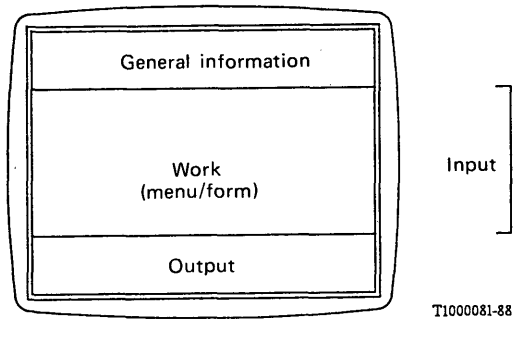


FIGURE B-3/Z.323

Dialogue Window

In this example, not all window areas are simultaneously visible. Work (menu/form) and input window areas are exclusive of each other. The user can replace one of these displayed window areas by the other one through the use of function keys.

B.5 *System status*

This window is used to display alarm indicators by an application managing exchange alarms. It contains two window areas:

- header;
- status.

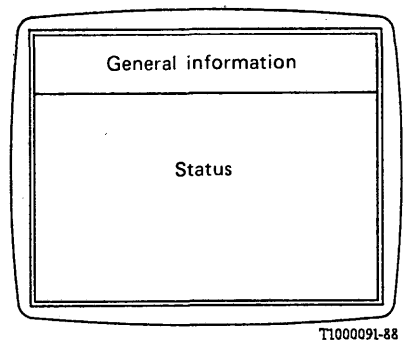


FIGURE B-4/Z.323

System Status Window

SECTION 4

SPECIFICATION OF THE MAN-MACHINE INTERFACE

Recommendation Z.331

INTRODUCTION TO THE SPECIFICATION OF THE MAN-MACHINE INTERFACE

1 Scope of the Section

The man-machine interface comprises the set of inputs, outputs and special actions, together with the man-machine interaction mechanisms, including dialogue procedures. Those elements are combined to manipulate the varied telecommunications functions which cover the management of SPC telecommunications systems. Consideration of these functions has been an essential prerequisite for the development of the CCITT MML Recommendations.

As stated in Recommendation Z.301, the CCITT MML can be used to facilitate operation, maintenance, installation, and acceptance testing of SPC systems. With the tendency of Administrations to centralize operations and maintenance jobs, many of the SPC systems functions may be controlled at terminals associated with operation and maintenance systems as well as at terminals associated with SPC systems. These terminals can be either local or remote relative to the system.

In order to help Administrations aiming to achieve uniformity among various systems, the MML Recommendations include not only the syntax of the language and dialogue procedures, but also the semantics relevant to the man-machine interface. Section 4 provides the means for deriving such semantics.

2 Organization of Section 4

Section 4 consists of the following Recommendations:

Z.331 Introduction to the specification of the man-machine interface

Z.332 Methodology for the specification of the man-machine interface – General working procedure

Z.333 Methodology for the specification of the man-machine interface – Tools and methods.

Z.334 Subscriber administration

Z.335 Routing administration

Z.336 Traffic measurement administration

Z.337 Network management administration

Recommendation Z.331 lists the operation, maintenance, installation and acceptance testing functions to be controlled by means of the MML.

Recommendation Z.332 presents the first part, the general working procedure of a methodology by which the man-machine interface can be generated for a particular functional area or subarea.

Recommendation Z.333 presents the second part, the tools and methods, of a methodology by which the man-machine interface can be generated for a particular functional area or subarea.

Recommendations Z.334 to Z.337 are based on the applications of phases 1, 2 and 3 of the methodology defined in Z.332 and Z.333 for the subscriber administration, routing administration, traffic measurement administration and the network management administration.

The main part of each Recommendation contains the model of the functional area or sub-area. Annex A of each Recommendation contains the list of functions to be controlled by means of the MML and the list of jobs considered in the development of the model. Annex B of each Recommendation contains a list of MML functions and associated information structure diagrams to be used as guidelines.

3 Functions to be controlled by means of the MML

The functions to be controlled by means of the MML are subdivided into four main areas: operation, maintenance, installation and acceptance testing. They are listed below. Based on the relationships existing among them, in each main area functions are grouped into functional areas and sometimes functional sub-areas. Due to the potentially different organization needs and system design philosophies, it is recognized that not all functions apply to every system.

The list of functions is not complete and it is expected to continue to evolve.

In particular, the issuing of Recommendations on specific areas or sub-areas will lead to the refinement of the preliminary list identified in this Recommendation for those functional areas or sub-areas. So far, this refinement has been achieved for subscriber administrations, traffic measurement administration, routing administration, network management administration (partially) specified in Recommendations from Z.334 to Z.337.

3.1 *Operation functions*

3.1.1 *Subscriber administration*[1] (see Recommendation Z.334)

- administering subscriber lines related data;
- tracing malicious calls;
- retrieving subscriber charging information;
- observing subscriber charging.

3.1.2 *Routing and digit analysis administration*

3.1.2.1 *Routing administration* (see Recommendation Z.335)

- managing the routing data base;
- querying the routing data base.

3.1.2.2 *Digit analysis administration*

- managing the digit analysis data;
- querying the digit analysis data base.

3.1.3 *Traffic administration*

3.1.3.1 *Traffic measurements administration* (see Recommendations E.502 and Z.336)

- performing traffic measurements;
- scheduling the execution of traffic measurements and the output of results;
- managing measurements data;
- retrieving measurements data.

---

[1] Subscriber administration deals with both single-line and multi-line subscribers.

3.1.3.2 *Traffic analysis administration* (see Recommendation E.502)

- inputting measured data;
- inputting the identification and capacity information of the measurement object;
- managing traffic data records;
- managing the output of reports;
- managing analysis description data;
- supervising the control of the time-delay of the various analysis operations.

3.1.4 *Tariff and charging administration*

- changing the tariff for a certain traffic destination;
- changing parameters for a charging rate;
- changing time for switching between day and night rate;
- reading accounting statistics (accounting between operating companies);
- changing the parameters involved in the accounting methods for traffic between different operating companies;
- retrieving of charging information.

3.1.5 *System control operation*

- setting and reading of a calendar;
- administering output routing;
- administering files;
- administering man-machine terminal capabilities;
- administering the system (hardware/software) configuration.

3.1.6 *User-system access control administration* (see Appendix I to Z.331)

- administering authority;
- retrieving authority information.

3.1.7 *Network management administration* (see Recommendation Z.337)

- performing measurements of network status and performance;
- performing network management actions;
- performing network management information distribution.

3.2 *Maintenance functions*

3.2.1 *Maintenance of subscribers' lines*

- testing one subscriber's line and associated equipment;
- testing a group of subscribers' lines and associated equipment;
- measuring one subscriber's line and associated equipment;
- measuring a group of subscribers' lines and associated equipment;
- blocking or unblocking a subscriber's line for maintenance purposes;
- observing or supervising of subscribers' lines and equipment.

3.2.2 *Maintenance of circuits between exchanges and associated equipment* (see Recommendation M.250)

- testing/measuring one circuit or a group of circuits and associated equipment;
- observing and supervising circuits and associated equipment;
- control the status of a circuit or a group of circuits and associated equipment;
- analysing maintenance data;
- administering and controlling maintenance reports.

3.2.3 *Switching network maintenance*

- making test calls;
- initiating a call trace;
- holding faulty connections;
- testing and measuring peripheral equipment (relay sets, signalling receivers and senders, etc.);
- testing and measuring switch units;
- reducing service for low priority subscribers;
- setting up a connection via a specific path through the network;
- supervising and measuring the quality of service of the switching network;
- localizing faults in the speech path network;
- providing access for traffic observation for maintenance purposes;
- reporting alarms;
- recording switch unit status.

3.2.4 *Control system maintenance*

- reporting system status;
- reporting alarms;
- localizing faults;
- testing on a functional basis after repair;
- initiating periodic testing operations;
- changing system configuration for maintenance purposes;
- checking consistency of data;
- initiating restart;
- setting traps for programme fault tracing;
- changing memory contents;
- memory dumping for maintenance purposes;
- controlling overload parameters;
- changing the criteria for the recognition of degradation of service;
- reducing service for low-priority subscribers.

3.3 *Installation functions*[2)]

3.3.1 *SPC system installation*

3.3.1.1 *SPC system hardware installation*

Installing:

- network blocks;
- trunks;
- signalling equipment;
- test equipment;
- blocks of subscriber-circuits;
- interface equipment;
- control equipment;
- memory equipment;
- input/output devices.

[2)] Installation also covers the extensions or reductions of the system after it is placed into service.

3.3.1.2 *SPC system software installation*

Installing:

- operational packages;
- test programmes;
- statistics programmes;
- programmes patches;
- signalling systems programmes;
- services, facilities programmes;
- system data.

3.4 *Acceptance testing functions*

Acceptance testing functions include any additional functions beyond those presented above to aid the administrations when testing a system to check its compliance with the Administrations' specifications.

APPENDIX I

(to Recommendation Z.331)

User-system access control administration

I.1 *General*

This appendix has been developed in accordance to the methodology defined in Recommendations Z.332 and Z.333.

The main part of this appendix deals with the model of User-System Access Control Administration. A glossary of the terms used is also included.

The list of functions to be controlled and the list of jobs are contained in Annex A.

For each function to be controlled by means of MML, one or more functions can be derived and each of them can be described using the metalanguage defined in Recommendation Z.333 in order to detail the relevant information structure.

Annex B contains a list of MML functions and information structure diagrams associated to each of them to be used as guidelines.

I.2 *Introduction*

User-system access control (here and after access control) is provided within a system to restrict the input allowed to be entered in order to prevent unauthorized system modification and or viewing of information.

Access control is the system function which performs the control of the access to systems and their functions by the users.

Access control administration is defined as the administration of the access rights of the users.

This Recommendation mainly covers human beings as users.

Machine to machine access control administration is not covered by this appendix.

It is therefore recognized that this appendix will require further study within a wider scenario including the various aspects of access control (man-machine, machine-machine, etc.).

I.3 *Access control model*

I.3.1 *Introduction*

Access criteria are defined to be the attributes that characterize the access to the system.

Permissions are defined to be the rights granted to the user.

Authority is defined to be the relationship between the access criteria and the permissions.

The inputs submitted are accepted by the system, provided that the system has verified the authority to enter them.

I.3.2 *Model*

The main attributes (see Figure I-1/Z.331) which have been adopted to identify access criteria and permissions are the following (other attributes of the two categories can be adopted depending on the administration's needs):

a) for access criteria
   - user identity
   - terminal identity
   - time interval

b) for permissions
   - command class
   - command parameters
   - system identity
   - time interval

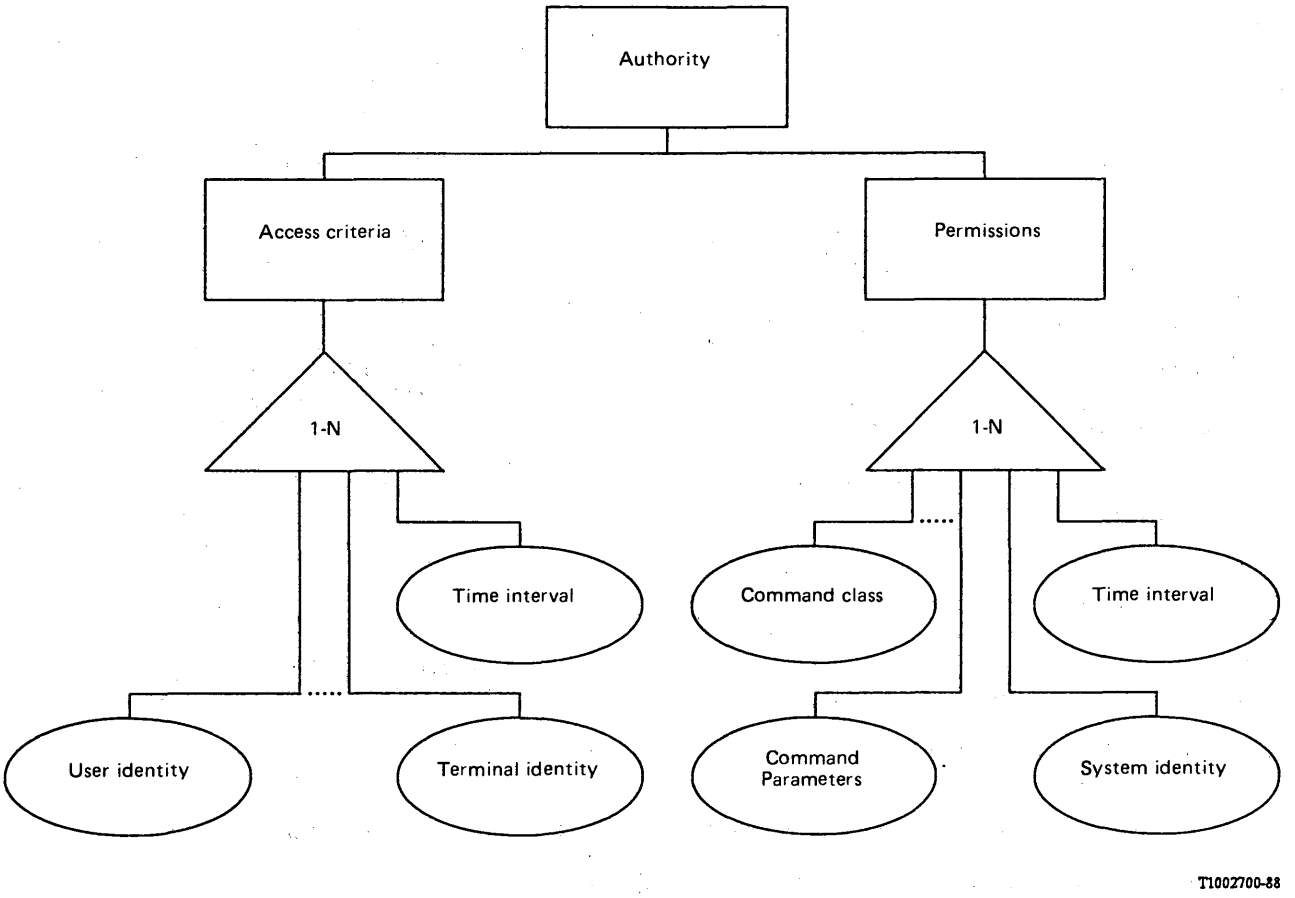


FIGURE I-1/Z.331

User-system access control administration

Some of the attributes listed above may not be implemented according to administration requirements.

In order to facilitate access control administration, groups may be formed in terms of single access control attributes (e.g. group of user identities can form a maintenance group).

An example of implementation is represented in Figure I-2/Z.331.

Authority					
Access criteria			Permissions		
User identity	Terminal identity	Time interval	System identity	Command class	Command parameters
User 1	Terminal 1	Any	Any	Any	Any
User 1	Terminal 2	8-17 h Monday through Friday	System 1	Subscrib. Administr.	Direct numb.: 81 000-82 000
User 2	Terminal 3	20-8 h	System 1	Junction maintenance	Junction identity 1A23 1800
User 3	Any	8-17 h	System 2	Subscrib. maintenance	Direct numb.: 73 000-87 000
Any	Terminal 4	8-17 h	Any	Subscrib. administr.	–
–	–	–	–	–	–

FIGURE I-2/Z.331

Example of application

I.3.3 *Attributes of access control*

In the following the meaning of the main attributes which are likely to be used in the access control administration, is described.

a) User identity

The user identity results from the identification procedure (see Recommendation Z.317) and uniquely identifies the user to the system.

In the identification procedure usually the identity of the individual user is used.

b) Terminal identity

The terminal identity is the identity of the I/O device as known to the system, via its hardware or logical connection.

c) Time interval

The access control may depend on the time when the input is entered and/or executed.

d) Command class

A command class can be either a single command code (see Recommendation Z.315) or an identifiable set of command codes.

e) System identity

System identity is the identity of the system or an application in which the command is allowed to be performed. In a centralized support system, individual systems connected to it may have their own access control. Alternatively, centralized control may be used based on the identity of the system addressed.

f) Command parameters

Access control may depend on a parameter (see Recommendation Z.315) or a combination of parameters. The control may be based on either the parameter name or the parameter name and its values.

If a parameter is considered, it may be desirable to limit such use to major objects in the system relevant to specific O&M Administration needs.

I.4 *Glossary of terms*

access criteria

The set of attributes that characterize the access to the system. Example attributes are user identity and terminal identity.

permissions

The rights granted to the user.

authority

The relationship between access criteria and permissions.

terminal identity

Identifies a physical terminal, a channel or a port to an SPC system.

I.5 *List of functions and jobs*

I.5.1 *List of system independent Class B functions*

I.5.1.1 Administering authority

I.5.1.2 Retrieving authority information

I.5.2 *List of jobs*

I.5.2.1 *To create/change authority*

- the purpose of the job is to create/change a specific authority by means of managing the relevant attributes;
- the system is supposed to record the data and check their correctness;
- the operator is supposed to input all needed data;
- the complexity of the job may be high depending on the amount of the data to be input;
- the frequency of the job is low.

I.5.2.2 *To delete a specific authority*

- the purpose of the job is to delete all the data related to the specific authority;
- the system is supposed to delete the data related to the authority;
- the operator is supposed to input the identity of the authority to be deleted;
- the complexity of the job is low;
- the frequency of the job is low.

I.5.2.3 *To interrogate the authority information*

- the purpose of the job is to retrieve authority information;
- the system is supposed to output the requested information on the selected device;
- the operator is supposed to input the identity of the access control attributes;
- the complexity of the job is low;
- the frequency of the job is low.

I.5.2.4 *To activate/deactivate an authority*

- the purpose of the job is to activate/deactive a specific authority previously created/changed; this job may be implied in the creation/changing job;
- the system is supposed to activate/deactivate the authority;
- the operator is supposed to input the date and the time for the activation/deactivation and the identity of the authority;
- the complexity of the job may be medium;
- the frequency of the job is low.

I.6 *Guidelines for the list of MML Functions and associated information structure diagrams*

I.6.1 *Introduction*

This section contains guidelines for the list of MML functions and associated structure diagrams related to the access control administration model defined in § 3 of this Recommendation.

I.6.2 *List of MML functions*

This list contains possible MML functions for the Access Control Administration.

This list is not mandatory nor complete; it may vary according to administration needs, telecommunication network levels, regulatory needs, etc.

I.6.2.1 *Creation*

- create authority

I.6.2.2 *Changing*

- change authority

I.6.2.3 *Deletion*

- delete authority

I.6.2.4 *Interrogation*

- interrogate authority

I.6.2.5 *Activation/deactivate*

- activate/deactive authority

I.6.3 *Information structure diagrams*

(To be developed.)

Recommendation Z.332

METHODOLOGY FOR THE SPECIFICATION OF THE MAN-MACHINE INTERFACE

GENERAL WORKING PROCEDURE

1 Introduction

Recommendation Z.331 provides a summary of the functions which are to be controlled by means of MML. Each functional area in this list is to be specified in detail to allow the generation of function-related semantics.

The use of such semantics in conjunction with the features provided by the Recommendations in Sections 2 and 3 allows the specification of the man-machine interface.

In order to produce a detailed specification, a formal method of working that provides a common approach is necessary. This Recommendation provides a methodology for such purposes.

In order to assign properly the responsibility for the application of the methodology, its application can be viewed as a two-stage process.

The first stage involves the generation of function-related semantics. This stage is aimed primarily at those experts working in CCITT Study Groups who are responsible for developing Recommendations associated with functions to be controlled by MML. However, it is recognized that the repertoire of such functions considered in CCITT Recommendations cannot cover the requirements of all Administrations or of all SPC systems. Therefore this stage is also aimed at Administrations, private operating agencies and scientific/industrial organizations who may find it necessary to specify functions peculiar to their individual needs.

The second stage of the application of the methodology involves the derivation of the actual man-machine interface using the semantics and the relevant features of Sections 2 and 3. This stage is the responsibility of Administrations, private operating agencies, and scientific/industrial organizations.

2 Orientation of the methodology: Administration centred and system centred

The methodolgy for the specification of the man-machine interface must be based on a common understanding of the concept of function.

Three different classes of system functions may be defined as follows:

1) *Class A functions or man-machine language (MML) functions*

   Those system functions which provide the MML user with the means of control of other system functions. The word "control" is assumed to include all types of inputs and outputs.

   Any Class A function can be subdivided into a general part which relates to e.g. the syntax check, information transmission control, etc., and an application part which relates tc the job in hand.

   Example: Create a traffic measurement.

2) *Class B functions*

   Those system functions which can be controlled at least partially by the MML user by means of MML functions.

   Example: Performing measurements of traffic parameters.

3) *Class C functions*

   Those system functions which are not at all controlled by the MML user in a given system during operation. Class C functions are not referred to in the following methodology.

The relationship between the concepts of "job" and the different types of functions is shown in Figure 1/Z.332.

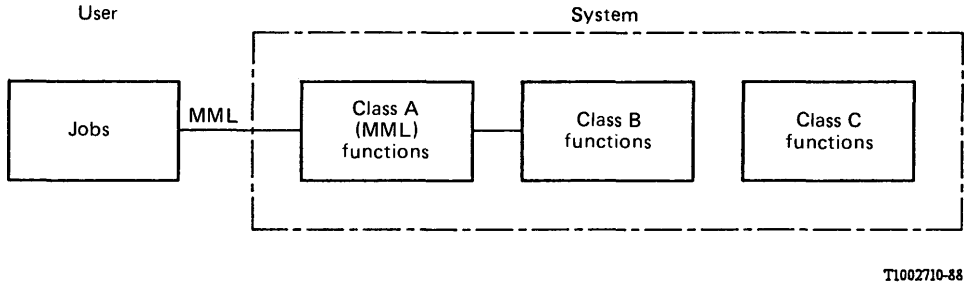


FIGURE 1/Z.332

This definition of MML function embodies the concept of both system actions and human actions performed on objects. The methodology presented in the following sections is based on the understanding of this concept.

To clarify the concept of "job" as applicable to operations and maintenance the following definition is provided.

Job

A discrete administrative activity within a telecommunications business which is designated as a part of the overall plan for running the business and characterized by man-machine communication and/or manual actions.

It is recognized that in the future the degree of automation of operation and maintenance jobs in the telecommunication network will increase as the application of auxiliary systems is broadened. Consequently, it is expected that all or part of a certain Class B function implemented in one system may appear as a Class C function in another system. The result is that the number and type of Class A functions supporting the same set of operation and maintenance jobs may differ from one system compared to another system.

3 General working procedure

The general working procedure consists of five phases:

1) the identification of Administration needs;
2) the identification, in sufficient detail, of the MML functions, i.e. those needed for control of the system by the user;
3) the identification of the information structure associated with each MML function;
4) the specification of the actual man-machine interface;
5) the verification and validation of phases 2, 3 and 4.

A more formal representation of this general working procedure is presented in Figures 2/Z.332, 3/Z.332 and 4/Z.332. The representation is made by means of Functional Block Interaction Diagrams as defined in the Z.100 series Recommendations on the Specification and Description Language (SDL). Figure 2/Z.332 represents the procedure at a high level showing its basic factors. Figure 3/Z.332 describes, at a lower degree of detail, the five phases presented above in terms of the information which should be produced and considered in each phase and their relationships. Figure 4/Z.332 describes, in the same terms, the two sub-phases into which phase 2 is further decomposed. As a drawing convention, information which is used primarily to support the activities performed in the various phases is indicated in the upper part of the Functional Block symbol.

Each phase is more fully described in the following paragraphs regarding its purpose, input and output products, relevant methods and tools, and CCITT Study Group responsibilities.

To achieve greater commonality among the various functional areas when performing phases 1, 2 and 3, harmonization of the terminology used is essential. A glossary of terms that may be useful in a number of functional areas has been provided in Recommendation Z.333.

This glossary is expected to evolve as MML function semantics activity continues. In addition, a glossary of terms specific to each functional area should also be provided as indicated below.

It should be emphasized that terminology harmonization refers to those phases of the methodology described herein which are the responsibility of the CCITT. It is not the intention of this recommendation, through its glossary or annexed examples, to recommend specific terminology for use at the actual man-machine interface. The present intent is rather that manufacturers and Administrations utilize the *concepts*, as here defined, that this terminology represents. They will select their own terminology to represent these concepts as applicable to their needs in specifying the actual interface. A common understanding of the definitions of these concepts will improve the coherence of the set of CCITT Recommendations in MML function semantics, as well as facilitate discussion concerning the capabilities of different systems with respect to the same as well as different functional areas.

The output of each phase is to be listed in a series of documents based on the terminology of Figures 3/Z.332 and 4/Z.332.

Phases	*Name*
1	Document A – List of Class B Functions and List of Jobs
2.1	Document B – Function Models
2.2	Document C – List of MML Functions
3	Document D – Information Structure of each MML Function
4	Document E – Specification of the man-machine interface
5	Document F – Verification and Validation Results
1-5	Document G – Glossary of terms.

The application of the methodology to a specific functional area may vary. Documents A-G may be produced for the functional area as a whole or the functional area may be divided into sub-areas and each treated separately. The primary rationale for the approach selected should be the coherence and maintainability of the total set of documents prepared for the functional area. If the second approach is selected, its details, including an unambiguous description of the main area and the identified sub-areas, should be documented also.

3.1 *Phase 1: identification of needs*

Purpose

To identify the various Administration needs in order to prepare a list of jobs to be performed by means of man-machine communications and to prepare an agreed list of system independent functions which are expected to be controlled by means of the MML (Class B functions). Terminology harmonization is essential.

Input

Inputs to the process of identifying Class B functions arise from three sources. First, CCITT Study Groups can provide operations and maintenance models and lists of Class B functions which are embodied in those models.

Second, Administrations can provide information on the jobs by which their systems are operated and maintained. Some indication as to the relative importance or frequency might be helpful in the process of specifying the man-machine interface.

The third input is the current version of Recommendation Z.331.

Output

List Class B Functions and List of Jobs (Document A).

These functions and jobs could be performed at terminals associated with operations and maintenance systems or SPC systems. A certain set of these functions and jobs might be able to be performed only at terminals associated with operations and maintenance systems or only at terminals associated with SPC systems.

Tools and methods

It will be necessary to take into account the following:

- directives from other Study Group experts;
- guidelines, as described in Recommendation Z.333;
- terminology harmonization guidelines, as described in Recommendation Z.333.

Use of SDL is also recommended.

3.2 *Phase 2: MML function identification*

Purpose

To identify, using harmonized terminology, MML functions related to Class B functions. This phase is an iterative procedure involving the application of several tools to identify the list of MML functions, i.e. those functions that are described in sufficient detail to allow the derivation of the man-machine interface. A diagrammatical representation of this phase is shown in Figure 4/Z.332.

Input

List of Class B functions and a list of jobs, both obtained as output of phase 1.

Output

- List of MML functions.
- Other information (whenever applicable).

} Document C

3.2.1 *Sub-phase 2.1: modelling*

Purpose

To represent, using harmonized terminology, the various functions of those parts of telecommunications systems controlled by MML by means of models.

Input

List of Class B functions.

Output

- Description of Class B functions by means of models.
- Other information (whenever applicable).

} Document B

Tools and methods

- At present informal modelling is available and there exists a need to identify and develop a formal method of modelling. SDL could be used for parts of the modelling work.
- Terminology harmonization guidelines, as described in Recommendation Z.333.

3.2.2 *Sub-phase 2.2: MML function decomposition*

Purpose

To identify, using harmonized terminology, each MML function considering both the model and the defined list of jobs.

Input

- List of jobs.
- List of system independent Class B functions.

Output

- List of MML functions.
- Other information (whenever applicable).

} Document C

Tools and methods

- The use of SDL is applicable. In order to represent or derive the MML functions, the MML function decomposition method should be applied.
- Terminology harmonization guidelines, as described in Recommendation Z.333.

3.3 *Phase 3: information structure identification*

Purpose

To identify, using harmonized terminology, the information structure of each MML function in order to provide a clear picture of the associated semantics (action, objects, information entities and their interrelationships). Separate diagrams for the structure of information related to input functions and to those outputs whose significance is such that benefits would be gained by their standardization should be provided.

The content of information structure diagrams should be limited to information related to such semantics. Other information, such as information related to possible parameter values, if desired, may be listed separately or as footnotes.

A one-to-one correspondence between information structure diagrams produced in this phase and the associated commands and outputs to be produced in Phase 4 is not implied. More specifically, a single information structure diagram could lead to a multiplicity of inputs or outputs. Also, several information structure diagrams could lead to a single input or output. Additionally, information structure diagrams should not be interpreted as a specification of any software process required to implement the related inputs and outputs.

Input

List of MML functions.

Output

- Information Structure Diagrams of each MML function.
- Additional information (a list of possible parameter values associated with Information Structure Diagrams).

} Document D

Tools and methods

Each MML function derived in phase 2 is in essence an action upon an object (or set of objects). An Information Structure meta-language is used to produce the Information Structure Diagrams associated to each MML function, as described in Recommendation Z.333.

Terminology harmonization guidelines, as described in Recommendation Z.333.

3.4 *Phase 4: specification of the actual man-machine interface*

Purpose

To present each input and output as it might appear on a man-machine communication terminal in terms of the related syntactic structure and to identify any related special actions. Also to select the appropriate dialogue procedures related to the MML functions.

The definition of inputs and outputs should be based on the type of interface to be derived, i.e. based on basic MML, or on extended MML or on both. In the latter case the consistency among commands and associated parameters should be pursued. The definition of inputs and outputs for an interface based on extended MML comprises the definition of menus and forms. This task should be achieved using the guidelines for the design of menus and forms contained in Recommendation Z.323.

Input

- The information structure representation of each MML function.
- Additional information.

Output

- Specification of the man-machine interface:
  a) inputs
  b) outputs
  c) special actions
  d) dialogue procedures
  e) interrelationships among a) to d).

Tools and methods

- The structure of inputs, outputs or special actions can be identified using guidelines as described in Recommendation Z.323, Z.333.
- A formal method to describe the syntactic structure of each MML input and output is given in Recommendation Z.333.
- Recommendations Z.302, Z.314-Z.317, Z.323.
- The use of SDL to describe the interactive operating sequences is recommended.

Note – Z.300-Series Recommendations do not deal with phase 4.

3.5 *Phase 5: verification and validation*

Purpose

To verify whether the MML functions identified previously together with their associated information structure lead to suitable procedures by which the users' needs can be satisfied.

To verify whether the man-machine interface identified in phase 4 leads to suitable procedures.

Input

- Information structure representations of each MML function.
- Preliminary man-machine interface.

Output

- An evaluation of the MML functions and their associated information structure.
- An evaluation of the preliminary man-machine interface.

} Document F

Tools and methods

- Procedure description method.
- Guidelines as described in Recommendation Z.333.

Note – Z.300-Series Recommendations do not deal with phase 5.

3.6 *Tools and methods*

Many tools and methods are available to provide assistance in reaching the goal of each phase described above. The applicability of each tool and method to a particular phase is dependent on the function being analysed. These tools and methods are described in Recommendation Z.333.

Examples of the use and application of these tools and methods for specifying functions are also included in Recommendation Z.333 and the Annexes to these Recommendations.

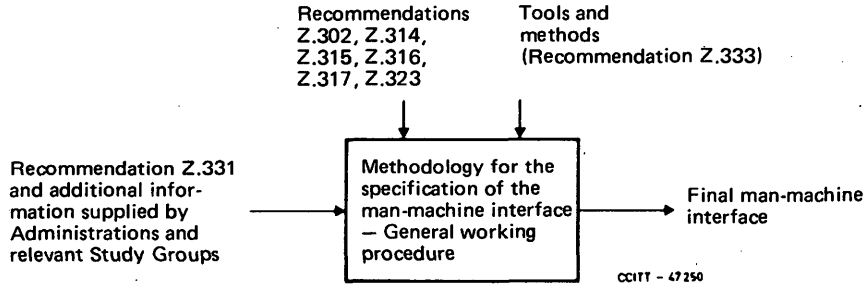


FIGURE 2/Z.332

High level view of the general working procedure of the methodology for the specification of the man-machine interface

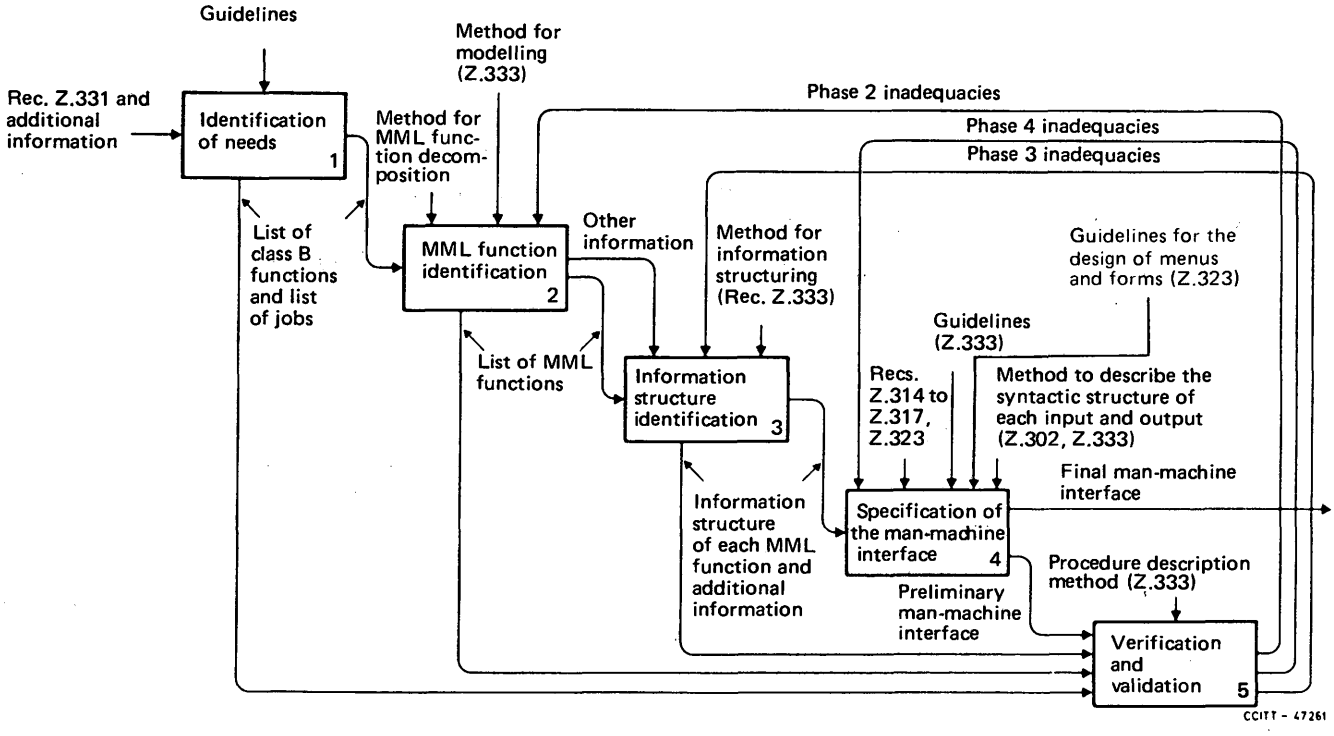


FIGURE 3/Z.332

General working procedure of the methodology for the specification of the man-machine interface

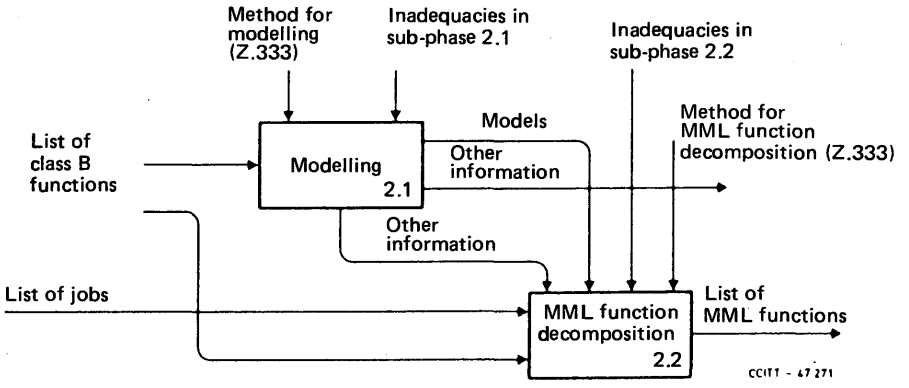


FIGURE 4/Z.332

Phase 2 of the general working procedure of the methodology for the specification of the man-machine interface

Recommendation Z.333

METHODOLOGY FOR THE SPECIFICATION OF THE MAN-MACHINE INTERFACE TOOLS AND METHODS

1 Introduction

This Recommendation presents the tools and methods that support the general working procedure described in Recommendation Z.332. Taken together, Recommendations Z.332 and Z.333 constitute the methodology for the specification of the man-machine interface.

2 List of tools and methods[1)]

The following tools and methods are necessary to support the methodology for the specification of MML functions:

- guidelines,
- modelling,
- MML function decomposition method,
- information structure metalanguage,
- procedure description method,
- formal representation of the syntactic structure of each input and output.

3 Description of available tools

3.1 *Guidelines*

3.1.1 *For phase 1*

Determine for each job:

- the purpose of the job;
- what the system is supposed to do;
- what the user is supposed to do;
- the complexity of the job from the users' perspective (see Note);

[1)] The tools and methods may be improved on the basis of user experience leading to additions or revisions.

– the frequency of the job (see Note);
– at which level in the network hierarchy the job is supposed to be performed (exchange, OMC);
– safety aspects.

Note – The following assumptions have been taken to better identify what is meant for "frequency" and "complexity" of a job.

3.1.1.1 *Frequency*

Low:

– if the job is supposed to be performed weekly or at longer intervals.

Medium:

– if the job is supposed to be performed daily.

High:

– if the job is supposed to be performed several times in a day.

3.1.1.2 *Complexity*

Low:

– low number of parameters (in general sense) – max 0 : 3;
– most information associated with these parameters are not compound;
– there is no semantic relationship among different parameters and parameter values.

Medium:

– the number of parameters is greater than 4 but less than 6-8;
– much information associated to these parameters is compound;
– there is no semantic relationship among parameters and/or parameter values.

High:

– there are many parameters;
– most information associated to these parameters is compound;
– there are semantic relationships among parameters and/or parameter values.

3.1.2 *For Phase 4*

No specific guidelines are provided for phase 2.

3.1.3 *For phase 3*

Three main categories of outputs can be identified within the MML function semantics specification, namely:

1) Response outputs inside the dialogue to the operator inputs.
2) Result outputs whose end-user is supposed to be the operator (e.g. results of reporting or interrogation functions).
3) Result outputs whose end-user is not assumed to be the operator (e.g. data collected for further elaborations).

The partitioning of the output media to be used and its component information entities should not be pursued in detail, with the following guidelines:

– Output media and output characteristics to support the first category of output (output inside dialogue) will not appear in the diagrams.
– Output media and output characteristics to support the second category will be as shown in Figure 1/Z.333.

It is also recognized that the lower level of detail, whose definition will depend on the individual Administration's needs, could in general include the information shown in Figure 2/Z.333.

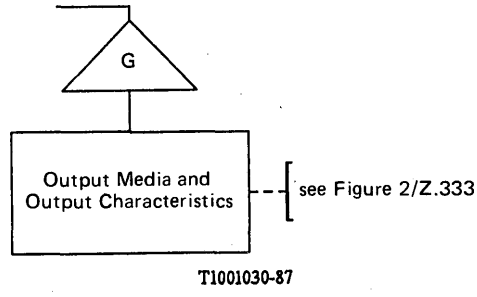


FIGURE 1/Z.333

Output media and output characteristics to support outputs whose end-user is supposed to be the operator

Output media to support outputs belonging to the third category will be shown if possible in the same way as the previous point.

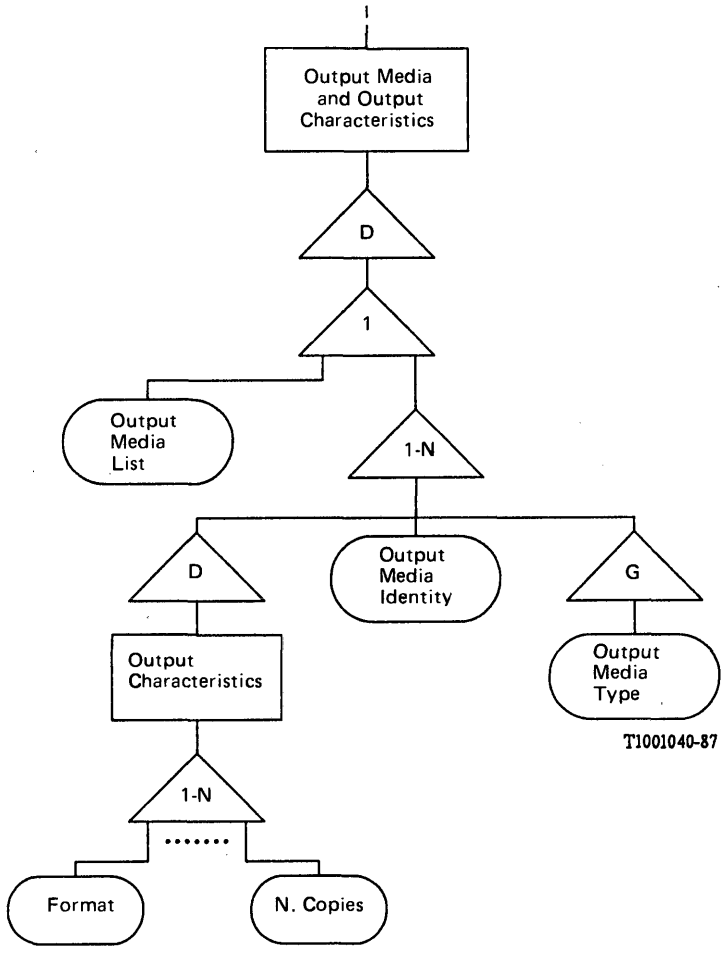


FIGURE 2/Z.333

Output media and output characteristics diagrams

3.1.4 *For phase 4*

To define individual menus and forms, follow the guidelines for the design of menus and forms as defined in Recommendation Z.323.

To define the individual inputs and outputs:

1) Consider what the system is supposed to do.
2) Select options in the function information structure.
3) Define the information to be represented by the command code or equivalent.
4) Define the information to be represented by the parameters and, if appropriate, their order.
5) For each parameter, when appropriate, identify the:
   - range of values,
   - default values,
   - information to be automatically supplied by the system.
6) Define the response outputs within dialogue, the interaction request outputs and outputs outside dialogue when applicable after considering the various mode operating sequences and the users' reactions to the outputs.
7) Define the associated syntactic structure.
8) Select terms and abbreviations for inputs and outputs.

3.1.5 *For phase 5*

1) Define a preliminary operational procedure in functional terms.
2) Finalize operational procedures.

3.1.6 *General guidelines*

1) Determine that the MML functions support the jobs to be performed.
2) It will be necessary to consider:
   - human factor aspects;
   - adequate allocation of authorities;
   - adequate definition of responsibility;
   - training of the user.

3.1.7 *Terminology harmonization guidelines for Phases 1-3*

To harmonize terminology:

1) Utilize existing CCITT vocabulary.
2) Select appropriate terms included in the general functional terminology (Appendix I).
3) Derive specific terms and their definitions pertinent to the functional area involved based on the following considerations:
   - common usage;
   - specificity;
   - translatability.

3.2 *Modelling*

Modelling involves the use of description text and/or figures drawn either with the support of formal symbology and rules (formal modelling) or without such rules (informal modelling).

3.2.1 *The need for models*

A tool available is the construction of informal models of those parts of telecommunications systems which have been selected for MML control. Also the organization of the Administration could be subject to modelling. Several models could apply when defining a job or an MML function. The use of models has the following advantages.

1) Models provide a means for the exchange of functional descriptions.
2) The validity of the derived man-machine interface can be consistently demonstrated by reference to the relevant models.

3.2.2 *Interpretation of models*

A model can be defined as an abstraction of a reality as seen from a certain viewpoint.

In Z.300 Recommendations the viewpoint assumed is that of their users, i.e. Administration specifiers and suppliers designers.

Models should therefore be interpreted as high level specifications and are not aimed to represent, suggest or imply any particular implementation.

They intend to provide only an overview in a conceptual sense of the information which is primarily relevant for the control of each particular functional area and of the main relationships among the various entities in the operator perspective.

Models produced expressly to determine the MML control structure are interpreted purely with that use in mind. Other models must lend themselves to the generation of MML control message sequences. CCITT feel bound to produce models which can be linked with the methods for determining information structure of MML functions.

3.3 *MML function decomposition*

The general MML functions are structured into component MML functions. Multiple levels of decomposition are allowed. For examples see the Annexes to this Recommendation.

3.4 *Information structure meta-language*

Each MML function identified at the lowest level of the MML function decomposition is structured into the information components needed to perform it. A top-down structuring is performed and multiple levels of information decomposition are allowed. The supporting tool is the meta-language presented below.

An aid in understanding of information structuring is to view a MML function as an action on an object(s). Information composed may therefore relate either to objects or to actions.

A general action associated with a MML function can be decomposed into subsidiary actions and modifiers to those actions. It is possible that no decomposition will take place. However, if decomposition is necessary, it should be noted that "decomposition" with respect to actions means both determining subsidiary actions *and* determining any qualifiers (modifiers, options, etc.) associated with the action. The latter is not a true decomposition.

3.4.1 *Decomposition meta-language*

3.4.1.1 *General*

The representation of the information structure associated with a MML function involves the specification of all needed information entities and their inter-relationships.

This representation can be achieved in a consistent manner by means of Information Structure Diagrams, drawn using the meta-language described below. Such meta-language consists of a set of symbols and drawing conventions.

A diagram represents the information structure in a top-down approach, starting from the identification of the MML function to be structured and ending with all the information components felt necessary in the man-machine interworking for that function.

The decomposition process is performed by the use of *sequences*, *selections* and *iterations*, by means of which any type of structure can be obtained.

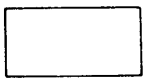
Unless otherwise stated, the sequence of information is not implied by the order in which different elements are presented in the diagram.

3.4.1.2 *Information entities*

3.4.1.2.1 *Composite parts*

A composite part is an information entity that can be further structured into smaller parts.

The following symbol is used:



3.4.1.2.2 *Component*

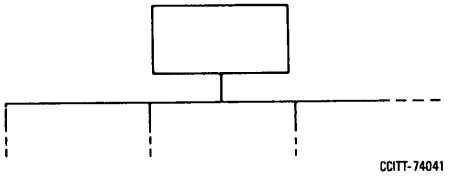
A component is an information entity that is not structured further.

The following symbol is used:

3.4.1.3 *Structuring*

3.4.1.3.1 *Subdivision*

Subdivision in Information Structure Diagrams is shown in the following way:

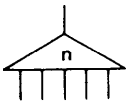


3.4.1.3.2 *Sequence*

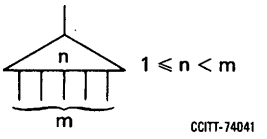
When the order between information entities is relevant, these are specified in sequence. A left-to-right sequence is indicated by the use of arrowheads as follows:

3.4.1.3.3 *Selection*

When a composite part is structured into a number of information entities, of which one or only some are relevant in any one case, a selection mechanism is used, represented by the following symbol:



In the general selection case, m possibilities exist from which selection must be made. Of these m possibilities a *specified* number, n, is to be selected, which implies $n < m$.



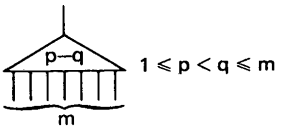
The number of possibilities to be selected, n, is given explicitly within the selection symbol, while the total number of possibilities, m, is given implicitly by the number of outlets from the selection symbol.

The following cases are allowed:

$n = 1$, $m > 1$ This is the most common selection case implying that one and only one of the possibilities is to be selected.

$n > 1$, $m > n$ Multiple selection of n of m possibilities.

If the number of choices to be made are variable between specified lower and upper limits, a number of possibilities are implied. In this case, both limits are given in the selection symbol:

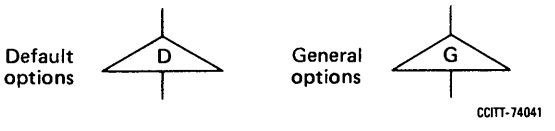


The lower limit p indicates the smallest number and q the largest number of *different* choices to be made out of the m possibilities. It should be noted that each choice may be selected only once.

3.4.1.3.4 *Options*

In some cases, options may be required such as default options or general options.

In these cases, the type of option is indicated by the appropriate capital letter only within the selection symbol, i.e. D for default options, G for general options. Only one outlet from the symbol is allowed:

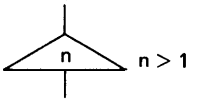


The use of a default option implies that the value taken by an information entity will be provided automatically if the user does not supply a value in the input.

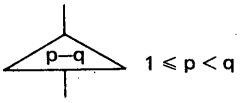
A general option is to be used for various reasons reflecting the needs of manufacturers and the needs of Administrations. The information entities that can be deduced from the outlet of this box can optionally be part of the man-machine interworking. This means either that the information exists in the system in a predetermined manner or that it is not needed. If this distinction must be made an annotation to the information structure diagrams should be made.

3.4.1.3.5 *Iteration*

When a composite part is structured into a number of information entities that can be repeated an arbitrary number of times, an iteration mechanism is used, represented by the following symbol, which has only one outlet:



If a number of interactions can vary within a range, the number of times a part is to be repeated is given as the lower limit p and the upper limit q.



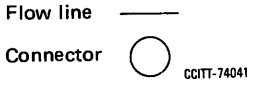
3.4.1.4 *Drawing conventions*

3.4.1.4.1 *Flow lines and connectors*

Every symbol is connected to the symbol it follows by a solid flow line.

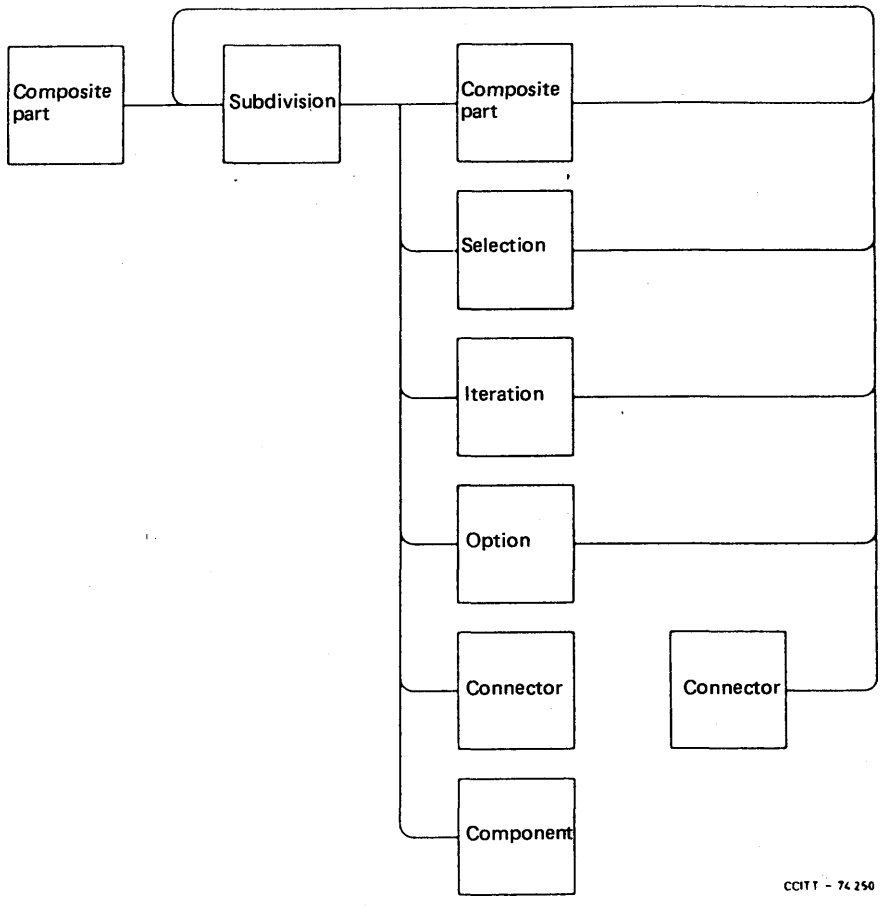
A solid flow line may be broken by a pair of associated connectors, with the flow assumed to be from the out-connector to its associated in-connector. Several out-connectors can be associated with the same in-connector.

Crossed flow lines should be avoided wherever possible.



3.4.1.4.2 *Connectivity rules*

Each information structure diagram begins with a composite part symbol and each path of a diagram ends with a component symbol. The drawing of diagrams must follow the connectivity rules represented below.



Note 1 – The symbol types and possible subdivision of flow lines are shown in square boxes.

Note 2 – Subdivision includes the trivial case of single continuous flow line.

3.4.1.4.3 *Annotations*

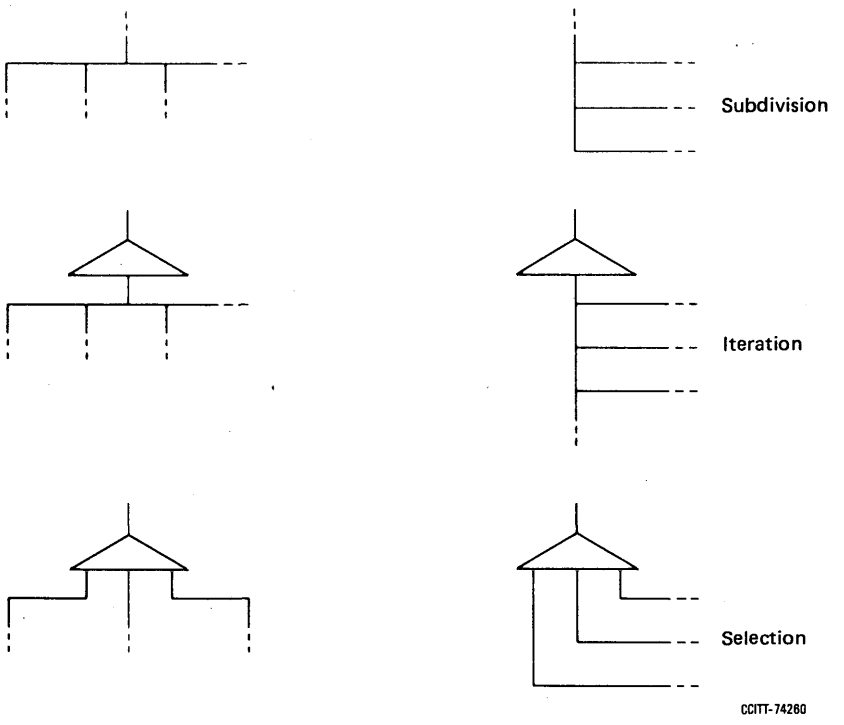
Annotations are denoted by the following symbol, where *n* is a number referencing a note providing descriptive and/or explanatory information.

Annotation – – – – – –[n

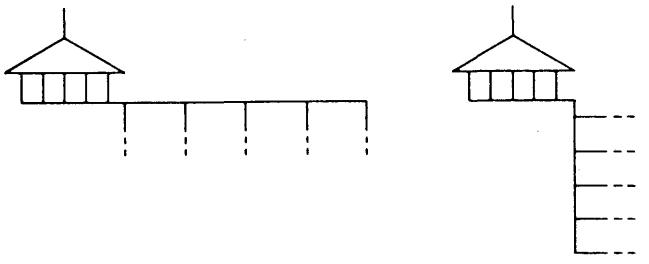
Annotations may be connected by a dashed line to any symbol or flow line.

3.4.1.4.4 *Special Notations*

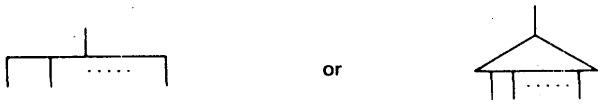
Instead of the normal structuring symbology where the structuring is shown horizontally, a vertical symbology may be used where this is advantageous, i.e. for saving of space. This vertical symbology may be used with all types of structuring.



For the selection symbol, in case of a high number of possibilities, the following drawing conventions are also allowed:



Where the number of information entities in a structure is undetermined, this could be shown as



depending on the type of structuring used.

3.5 *Procedure description method*

Man-machine dialogue may be considered a feature of an SPC system and may be represented by means of two processes: one related to the user, the other related to the system. These two processes exchange information by means of signals that for MML purposes are intended to be mostly inputs and outputs.

In particular, the description of MML operational procedures may be made by focusing the attention on one of the machine logic functions, the associated MML function, and describing the process that performs this function.

To reduce the complexity of drawings, it seems useful to limit the description to the main signals between user and system, i.e. inputs and outputs, and to omit showing features such as timing, error reporting, editing procedures, etc., that may be described elsewhere by means of SDL if needed. For an example see Appendix II.

3.5.1 *Features to be used in the description*

A MML operational procedure can be considered as a process whose behaviour may be specified in terms of inputs, states, transitions, decisions, outputs and tasks.

In the following paragraphs, basic concepts of SDL are interpreted in the context of MML applications.

3.5.1.1 *Input*

An input is a set of data which is input by the user and which is recognized by the MML operational procedure. Input may be, e.g commands in direct information entry, or other types of data.

3.5.1.2 *State*

A state is a condition in which the action of the MML procedure is suspended awaiting an input.

3.5.1.3 *Transition*

A transition is a sequence of actions which occurs when a MML operational procedure changes from one state to another as a reaction to an input.

3.5.1.4 *Decision*

A decision is an action within a transition which asks a question to which the answer can be obtained at that instant and chooses one of several possible paths to continue the transitions.

3.5.1.5 *Output*

Output is a set of data which is output by the MML operational procedure and which in turn is used as an input to the operational process.

3.5.1.6 *Task*

A task is any action within a transition that is neither a decision nor an output.

3.5.1.7 *Symbols and rules*

Symbols and rules are those defined in the SDL Z.100 series Recommendations.

3.6 *Formal representation of the syntactic structure of specific inputs and outputs*

The formal representation of the syntactic structure of specific inputs and outputs may be provided by the use of the existing syntax metalanguage in Recommendation Z.302. The use of the Backus Naur Form (BNF) has also been suggested as possibly being more effective. As advanced terminal capabilities are being considered by the MML Sub-Working Party, additional methods may be needed. The suitability of these methods must be studied further, and if possible, a single method recommended.

3.6.1 *Backus Naur Form (BNF)*

Inputs and outputs are defined as sequences of terminal elements and/or non-terminal elements.

Terminal elements are characters belonging to the MML character set as defined in Recommendation Z.314 and the syntactic elements as defined in Recommendations Z.314, Z.315 and Z.316. Syntactic elements are indicated by means of their name written with small letters between angular brackets (< and >).

Non-terminal elements are elements that must be further defined again as sequences of terminal and/or non-terminal elements. They are indicated by one or more words written with small letters between angular brackets (< and >).

3.6.1.1 *Notation*

Definitions are indicated by writing commands or non-terminal elements on the left hand of the symbol ::= (double colon, equal sign) and, on the right hand side, one or more sequences of terminal and/or non-terminal elements.

Alternative choices are indicated separated by | (vertical bar).

Terminal and non-terminal elements may be grouped together by using braces ({ and }). Repetition of these groups is indicated by means of two subscripts after the braces, one for the minimum, one for the maximum number of times the group may be repeated.

If a group of terminal and non-terminal elements is written between brackets ([and]) the group is optional.

For an example see Appendix III.

APPENDIX I

(to Recommendation Z.333)

Glossary of common terms used in the specification of the man-machine interface

This glossary of common terms is to be utilized where applicable by CCITT bodies when applying phases 1-3 of the methodology. It is expected to evolve as the methodology is applied to a wider range of areas. This document is not intended to constrain manufacturers' and administrations' choice of terms to represent these concepts at the actual man-machine interface.

It has been noted in Recommendation Z.332 that it is useful to view MML functions as *actions* upon *objects*. The concepts represented by the terms in the present collection are limited to action concepts. It is expected that as this glossary evolves, most action concepts will be defined here since they generally apply to more than one functional areas. Conversely, object concepts will generally be specific to a functional area and thus defined in the glossary associated with a functional area.

Among the concepts for actions that may be performed at the man-machine interface are concepts for which the proper object of the action is:

- data only
- equipment only
- either data or equipment.

These three categories of actions correspond to the three major divisions of this glossary.

A number of the concepts below are best understood and normally utilized in complementary pairs; these cases will be indicated by notation such as CREATE/DELETE.

I.1 *Data management actions*

The term **data set** is defined to be a user-accessible set of one or more data items characterized by a particular use, and also by the constraints on data format and/or values that make it suitable for this use.

I.1.1 *CREATE/DELETE*

The following concepts concern user control of the existence of data sets within the system.

CREATE: Establish in the system a new data set.

Examples: CREATE A MEASUREMENT SET, CREATE AN OBJECT LIST.

DELETE: Eliminate a data set from the system.

Examples: DELETE A MEASUREMENT SET, DELETE AN OBJECT LIST.

I.1.2 *CHANGE and EDIT*

The modification of data is normally accomplished by one of two basic methods. The first method (CHANGE) is through the use of functionally specific inputs and outputs intended to be used to modify particular data set types or even particular data items within those data sets. The second method of data modification (EDIT) allows the user to perform changes directly to a display of the data that is to be modified.

Taking this into account, CCITT organizations applying the methodology described in this recommendation should employ the term CHANGE for any data modification requirements, except in cases where the capability to EDIT would have clear advantages, such as in the example given below.

CHANGE: Modify specified data items in a data set via an input or inputs intended for that purpose.

Example: CHANGE ANALYSIS THRESHOLDS.

EDIT: Display a specified data set and subsequently modify the data set. A common system capability, e.g., editor, is normally used to support such an action.

Example: EDIT TRAFFIC DATA RECORDS.

I.1.3 *ACTIVATE/DEACTIVATE*

The creation of a data set does not necessarily imply that that data is immediately available for use by the system for its intended purpose. The following concepts make a previously created data set available or unavailable to the system.

ACTIVATE: Initiate a system process that requires preliminary data entry, or make a previously entered data set available to the system for its intended use.

Examples: ACTIVATE A MEASUREMENT, ACTIVATE A ROUTINE TEST.

DEACTIVATE: Terminate a system process initiated by an ACTIVATE action, or make a data set unavailable for use by the system.

Examples: DEACTIVATE A MEASUREMENT, DEACTIVATE A ROUTINE TEST.

I.1.4 *FILTER and SORT*

These concepts allow the user to manipulate data to be subsequently accessed or stored.

FILTER: Form a subset of a data set consisting of all data items in the set meeting specified criteria. The original data set is unaffected by this action.

Example: FILTER TROUBLE OR RESTORAL REPORTS.

SORT: Rearrange the order of a data set according to specified (or default) criteria. The contents of the original set is not affected by this action, only its order.

Example: SORT A FILE OF NAMES (e.g. in alphabetical order).

I.1.5 *INTERROGATE and BROWSE*

The concepts below describe system actions that allow user access to specified portions of the data that has been created by the user or by the system.

INTERROGATE: Provide a display of the current values of the items in one or more data sets.

Examples: INTERROGATE A MEASUREMENT, INTERROGATE A MEASUREMENT TYPE.

BROWSE: Display sequentially the current values of items in a data set. The user may examine the data items in either the forward or backward direction.

Example: BROWSE REPORT FILES.

I.1.6 *INPUT/OUTPUT and ROUTE*

The concepts in this section concern the transfer of data from one location to another.

INPUT: Enter data by means of a user terminal into the system.

Example: INPUT TROUBLE OR RESTORAL REPORT.

OUTPUT: Transfer specified data from the system to the user terminal (e.g. VDT, printer).

Example: OUTPUT SUMMARY REPORT.

The distinction between OUTPUT and INTERROGATE (I.1.5) is that INTERROGATE simply gives a read-back of user-created data, whereas OUTPUT refers to data upon which the system itself has acted in some way, e.g. reports.

ROUTE: Instruct the system that any subsequent messages, classes of data, or message types indicated should be output to specified media.

Example: ROUTE OUTPUT OF REPORTS.

I.2 *Equipment Management Actions*

I.2.1 *REMOVE/RESTORE and SET*

Equipment units can often simply be placed either out of service or in service under software control. The pair REMOVE/RESTORE represents this pair of actions. Manipulation of the status of objects with a more complicated set of maintenance states is expressed by the system action SET, which normally also covers the out of service and in service states. The REMOVE/RESTORE pair is used frequently and is sufficient for a large range of equipment, hence is singled out here as an important special case of the SET action.

REMOVE: Take specified equipment units out of service. The system still retains knowledge of the units so that they may be returned to service by the RESTORE action defined below, automatic recovery, or manual override.

Example: REMOVE CIRCUIT.

RESTORE: Return specified units to service.

Example: RESTORE CIRCUIT.

SET: Place equipment in a specified state (number of states >2). Possible states include in service and out of service.

Example: SET EQUIPMENT UNIT.

I.2.2 *ALLOW/INHIBIT*

Modern systems (e.g. for maintenance or control) utilize many system functions which occur automatically or dependent only upon the detection of certain conditions. Often it is essential to be able to instruct the system *not* to perform these functions, even should the appropriate set of conditions arise. The complementary capability to return the automatically controlled function to its normal state must then also be provided.

ALLOW: Permit specified system actions, system responses or functions to occur. These functions may be inhibited by system design or by the INHIBIT system action defined below.

Example: ALLOW THRESHOLD.

INHIBIT: Prevent the specified system actions, system responses or functions from occurring. These functions may normally be allowed by the system design, or by the ALLOW action defined above.

Example: INHIBIT THRESHOLD.

I.3 *Management actions that may apply to data or equipment*

INITIALIZE: Put specified data or equipment into a predefined initial (normal) condition or value.

Examples: INITIALIZE THRESHOLD COUNTER, INITIALIZE OUTPUT DEVICE.

EXECUTE: Perform a predefined procedure.

VERIFY: Perform the enforcement of a consistency rule on a specified set of data.

CONNECT: Make a connection between two existing entities.

DISCONNECT: Break a previously established connection.

START: Initiate a procedure or process.

STOP: Terminate the specified activity and leave the system in a defined state.

SUSPEND: Hold an activity temporarily.

RESUME: Continue an activity previously suspended.

APPENDIX II

(to Recommendation Z.333)

Procedure description example

The job of "create a new traffic measurement" is described as a procedure in which two different SDL processes, the user process and the system process, are shown.

Only the relevant aspects of the procedure are represented in the diagrams; some features are omitted such as a rejection output due to syntactic errors and related correction procedures etc., which are common to the other procedures.

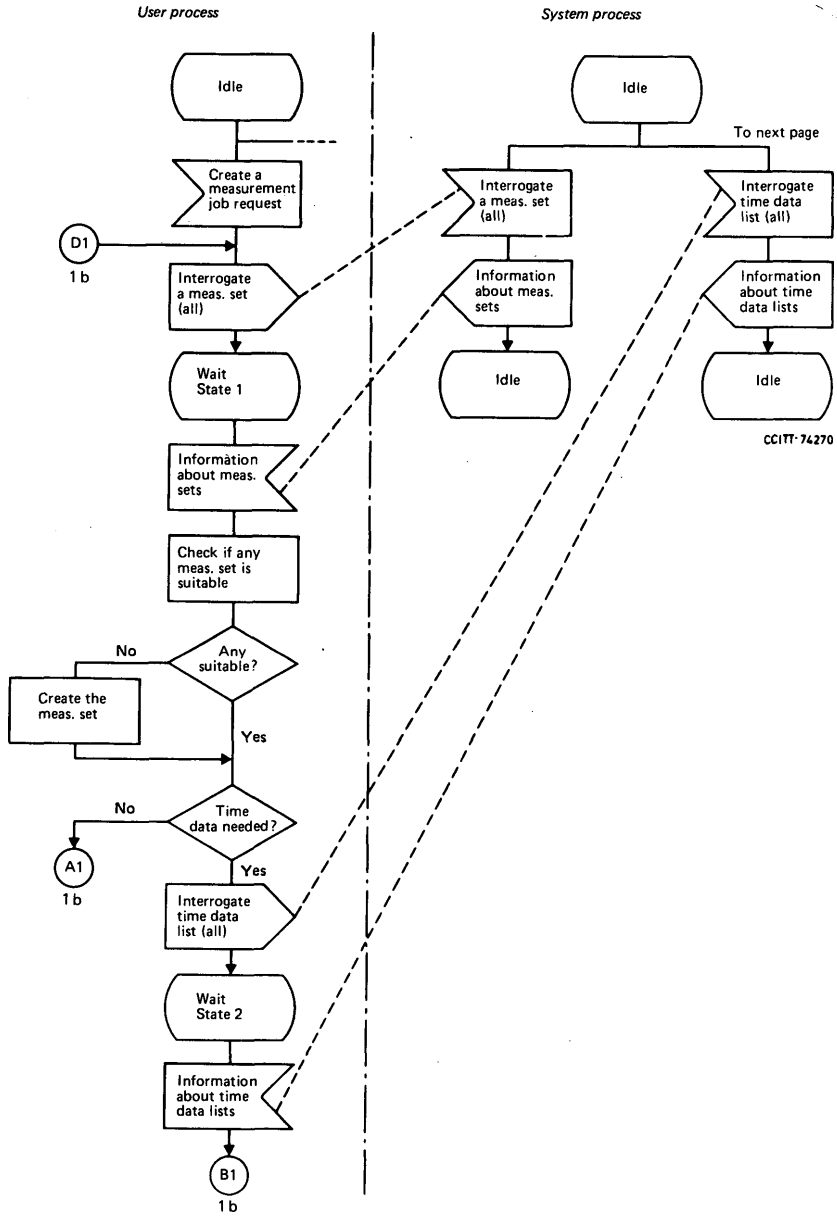


FIGURE II-1a/Z.333

Procedure description example

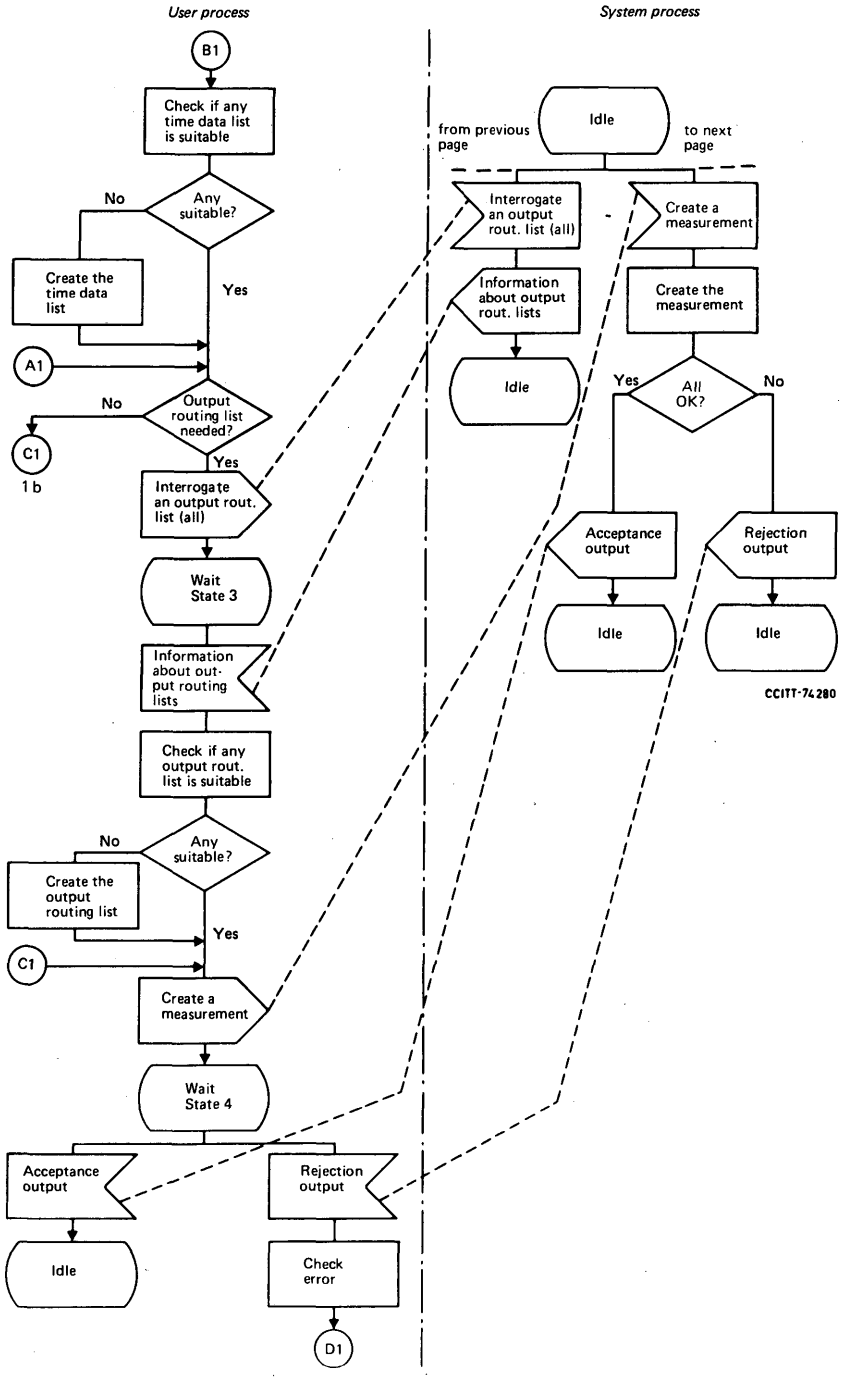


FIGURE II-1b/Z.333

Procedure description example *(cont.)*

APPENDIX III

(to Recommendation Z.333)

Examples of the use of the Backus Naur Form (BNF)

Applying the BNF meta-language described in § 2.6.1 to the traffic measurement functions (see Recommendation Z.336, Annex A, (Figures B-9/Z.336 and B-14/Z.336), the following BNF examples are derived with the assumption of a one to one relationship between the MML function and associated command:

a) *Function "create an object list":*

<create an object list> ::= <command code>:
<object list identity>
{,<list of objects of one type>}$^{1-N}$;

<object list identity> ::= <parameter name> = <symbolic name>

<list of object of one type> ::= <type of objects> = <objects identity>

<type of objects> ::= <parameter name>

<object identity> ::= <decimal numeral> {{&<decimal numeral>}|{&&<decimal numeral>}}$^{O-N}$|
<symbolic name> {&<symbolic name>}$^{O-N}$

b) *Function "delete an object list"*

<delete an object list> ::= <command code>:
<list of identities of object list>;

<list of identities of object list> ::= <parameter name> =
<symbolic name> {&<symbolic name>}

Recommendation Z.334

SUBSCRIBER ADMINISTRATION

1. *General*

This Recommendation has been developed in accordance with the methodology defined in Recommendations Z.332 and Z.333.

The main part of this Recommendation deals with the model of the subscriber Administration. A glossary of the terms used is also included. The list of functions to be controlled by means of MML is contained in Annex A. For each of these functions one or more MML functions can be derived and each of them can be described using the metalanguage defined in Recommendation Z.333 in order to detail the relevant information structure.

Annex B contains a list of MML functions and the information structure diagrams associated to each of them to be used as guidelines.

2 *Introduction*

The scope of this Recommendation is the operational aspect of the functional area of subscriber administration.

It has been recognized that subscriber Administration is related to various administrative procedures, belonging to other functional areas, which have to be carried out prior to or in functional relationship to the jobs to be controlled by MML.

Such administrative procedures are Administration dependent with different degrees of data processing support.

They can be considered as the administrative environment.

3 Subscriber administration model

3.1 *Introduction*

The model is applicable to subscribers to whom normal telephone services (POTS: Plan of Telephone Services) are offered.

For the representation of the model the metalanguage defined in Recommendation Z.333 has been used.

For the purposes of this Recommendation subscribers have been divided into the following two classes (see Figures B-1/Z.334, B-2/Z.334 and B-3/Z.334 :

- single-line subscribers, comprising single-party and multi-party lines;
- multi-line subscribers, comprising PBX (without direct inward dialling), PABX (with direct inward dialling) and subscriber line groups.

Both single-line and multi-line subscribers are embedded in an administrative environment. This environment may comprise all data relevant for administering the various types of subscribers. Examples for such data are addresses of subscribers and subscribers' features. The administrative environment aspects are not considered in detail in this model.

3.2 *Description of model*

3.2.1 *General*

Subscribers are distinguished by the data describing the way in which they are connected to and managed by the public exchange, including data related to the associated hardware equipment in the exchange. Examples are two party lines, ordinary subscriber lines, PABXs multi-party lines.

From the Administration view every subscriber falls into one of the two main types identified in the model, single- or multi-line subscribers. Due to the different association of information entities needed for the representation of these two main types, two different sets of MML functions are felt to be appropriate. The relevant information entities defined in the model are described in the following sections.

3.2.2 *Line characteristics*

Line characteristics are described by their attributes which may include information about class of line, kind of signalling, attenuation equalization, traffic directions, etc.

3.2.3 *Line group characteristics*

Line group characteristics are described by their attributes which may include information about kind of signalling, class of line, traffic directions etc., associated to all lines forming the group.

3.2.4 *Facility characteristics*

Facility characteristics are described by their attributes. They include all information about supplementary services which can be assigned to a given subscriber. Examples are abbreviated dialling, do not disturb service, wake-up service, charging information like normal charging or free of charge.

There are attributes which can only be controlled by the Administration while others are primarily controlled by the subscribers themselves. However, it has been recognized that in the latter case these attributes can also be controlled by the Administration.

3.2.5 *Restriction characteristics*

Restriction characteristics include information which indicate the limitations on the regular operational mode, e.g., traffic restrictions for originating calls.

3.2.6 *Monitoring characteristics*

Monitoring characteristics are related to particular monitoring actions, which are carried out by the system. The monitoring actions are mainly of a temporary nature and comprise subscriber Administration functions carried out to obtain data related to calls. Examples are malicious call tracing and charging observation. Monitoring actions are activated on subscriber's or Administration's request and are described by their attributes, e.g. duration of monitoring, starting time.

3.2.7 *Single-line subscriber identity*

Single-line subscriber identity allows the unambiguous identification of a single-line subscriber, normally by its directory number.

3.2.8 *Multi-line subscriber identity*

Multi-line subscriber identity allows the unambiguous identification of a multi-line subscriber, normally by its directory numbers.

3.2.9 *Equipment identity*

Equipment identity allows the unambiguous identification of the device to which a subscriber line is connected.

3.2.10 *Line number*

A line number allows the unambiguous identification of a line within a line group.

3.2.11 *Associated directory number*

An associated directory number may be entered for lines that are members of a multi-line subscriber and that need to have associated with them a directory number other than the multi-line subscriber identity. Examples include the night service number for a member of a PBX or the directory number of a directly dialable member of a multi-line hunt group.

3.2.12 *State*

The state of a subscriber line identifies the current operational mode in an unique way. It may be possible to interrogate the state of a subscriber line.

4 *Glossary of terms*

single-line subscriber line

A line between a public exchange and a subscriber set.

multi-line subscriber line

A line between a public exchange and a P(A)BX or a line between a public exchange and a subscriber set belonging to a subscriber line group.

line group

A line group is a group of lines of a multi-line subscriber with some common line characteristics, e.g. incoming, outgoing, bothway.

subscriber line group

A group of line groups which are recognized and managed by a public exchange as a logical group.

ANNEX A

(to Recommendation Z.334)

List of system functions to be controlled by MML and list of jobs

A.1 *List of functions to be controlled by MML*

1) administering subscriber lines and related data;
2) tracing malicious calls;
3) retrieving subscriber charging information;
4) observing subscriber charging.

A.2 *List of jobs*

A.2.1 *General*

The jobs considered hereafter can be performed either at the exchange level or at Operation and Maintenance Centre (OMC) level, or both.

The description of each job should include the following general characteristics.

- The operator is supposed to input all data relevant for the job to be performed.
- The system is supposed to check the input data for formal correctness and logical plausibility and to output an error message in the case of syntax/semantic errors and incomplete insertions, and to prompt for further input.
- The system should update the data in its database according to the requirements of the performed job, e.g. to store, to delete data in its database.

A.2.2 *List of jobs*

A.2.2.1 *To create a single-line subscriber*

- The purpose of the job is to define an association between an equipment identity, subscriber identity, line, facility, restriction and monitoring characteristics.
- The complexity of the job is medium.
- The frequency of the job is high. In local exchanges the job is expected to be one of the most frequent.
- The job is performed at the request of the subscriber, in accordance with the availability of required equipment, directory numbers, etc.

A.2.2.2 *To create a multi-line subscriber, a new line group of a multi-line subscriber, or a new line of a multi-line subscriber*

- The purpose of the job is to create a multi-line subscriber, a new line group of a multi-line subscriber, or a new line of a multi-line subscriber, and to define an association between multi-line subscriber identity, line group identity, line numbers, equipment identities, line, facility, restriction and monitoring characteristics.
- The complexity of the job is medium.
- The frequency of the job is low.
- The job is performed at the request of the subscriber, in accordance with the availability of required equipment, directory numbers, etc.

A.2.2.3 *To change single-line subscriber related data*

- The purpose of this job is to change single-line subscriber related data, i.e. the line and/or facility and/or the restriction and/or the monitoring characteristics.
- The complexity of the job depends on the number of the changes and assignments.
- The frequency of the job is medium.
- The job is performed at the request of the subscriber or of the Administration.

A.2.2.4 *To change multi-line subscriber related data*

- The purpose of the job is to change multi-line subscriber related data, i.e. the facility and/or line and/or restriction and/or monitoring characteristics and/or associated directory number.
- The complexity of the job depends on the number of the changes and assignments.
- The frequency of the job is low;
- The job is performed at the request of the subscriber or of the Administration.

A.2.2.5 *To delete a single-line subscriber*

- The purpose of the job is to delete all data, i.e., equipment identity, subscriber identity and characteristics, related to a certain single-line subscriber.
- The complexity of the job is low depending on system checks.
- The frequency of the job is medium.
- The job is performed on request of the subscriber or of the Administration.

A.2.2.6 *To delete a multi-line subscriber, line groups of a multi-line subscriber, or lines of a multi-line subscriber*

- The purpose of the job is to delete a multi-line subscriber, line groups of a multi-line subscriber or lines of a multi-line subscriber.
- The compexity of the job is medium depending on system checks.
- The frequency of the job is low.
- The job is performed at the request of the subscriber or of the Administration.

A.2.2.7 *To interrogate single-line or multi-line subscriber related data*

- The purpose of the job is to interrogate single-line or multi-line subscriber, line related data, according to selection criteria, e.g. single-line/multi-line subscriber identity, all charge free lines.
- The system is supposed to display the desired data on an output device at the operator's request.
- The complexity of the job is low.
- The frequency of the job is high when the selection criterion is subscriber identity and low when other selection criteria are used.
- The job is performed at the request of the Administration.

A.2.2.8 *To retrieve charging information for a single-line or a multi-line subscriber*

- The purpose of the job is to retrieve charging information for a single-line or a multi-line subscriber in case of the pulse metering.
- The system is supposed to provide subscribers' charging information on an output device at the operators' request.
- The complexity of the job is low.
- The frequency of the job is low.
- The job is performed for administrative reasons.

A.2.2.9 *To block/unblock a single-line subscriber*

- The purpose of the job is to make a single-line subscriber unavailable/available to traffic.
- The system is supposed to block/unblock a single-line subscriber to originating and/or terminating traffic.
- The compexity of the job is low.
- The frequency of the job is medium.
- The job is performed for administrative reasons.

A.2.2.10 *To block/unblock a multi-line subscriber*

- The purpose of the job is to make a multi-line subscriber, line groups of a multi-line subscriber or lines of a multi-line subscriber unavailable/available to traffic.
- The system is supposed to block/unblock a multi-line subscriber, line groups of a multi-line subscriber or lines of a multi-line subscriber to originating and/or terminating traffic.
- The complexity of the job is low.
- The frequency of the job is low.
- The job is performed for administrative reasons.

A.2.2.11 *To activate/deactivate malicious all tracing for a single-line or a multi-line subscriber*

- The purpose of the job is to enable/disable malicious call tracing for a single-line subscriber, a multi-line subscriber, or line groups of a multi-line subscriber.
- The system is supposed to collect the malicious call tracing data, to store it and to display it on operator's request.
- The complexity of the job is low.
- The frequency of the job is low.
- The job is performed at the request of the subscriber.

A.2.2.12 *To activate/deactivate a single-line or a multi-line subscriber*

- The purpose of the job is to put into or to take out of service a single-line subscriber, a multi-line subscriber, line groups of a multi-line subscriber or lines of a multi-line subscriber previously defined in the system. The activation function may be implied in the corresponding creation function.
- The complexity of the job is low.
- The frequency of the job is high.
- The job is performed at the request of the subscriber.

A.2.2.13 *To activate/deactivate single-line or multi-line subscriber charging observation*

- The purpose of the job is to start/stop charging observation for a single-line or multi-line subscriber for a stated duration.
- The system is supposed to collect the charging information data, to store it and to output it on operator's request.
- The complexity of the job is low.
- The frequency of the job is low.
- The job is performed at the request of the Administration.

ANNEX B

(to Recommendation Z.334)

Guidelines for the list of MML functions and associated information structure diagrams

B.1 *Introduction*

This annex contains guidelines for the list of MML functions and associated information structure diagrams related to the subscriber administration model defined in this Recommendation Z.334, § 3.

B.2 *List of MML functions*

The list contains possible MML functions for subscriber administration.

This list is not mandatory or complete. It may vary according to administrative needs, telecommunication network levels, regulatory needs, etc.

1) *Creation*

- create a single-line subscriber;
- create a multi-line subscriber, a new line group of a multi-line subscriber, or a new line of a multi-line subscriber.

2) *Change*

- change single-line subscriber related data;
- change multi-line subscriber related data.

3) *Deletion*

- delete a single-line subscriber;
- delete a multi-line subscriber, line groups of a multi-line subscriber or lines of a multi-line subscriber.

4) *Interrogation*

- interrogate single- or multi-line subscriber related data.

5) *Retrieval*

- retrieve charging information for a single-line or a multi-line subscriber.

6) *Block/unblock*

- block/unblock a single-line subscriber;
- block/unblock a multi-line subscriber.

7) *Activation/deactivation*

- activate/deactivate malicious call tracing for a single-line subscriber;
- activate/deactivate malicious call tracing for a multi-line subscriber;
- activate/deactivate single-line subscriber charging observation;
- activate/deactivate multi-line subscriber charging observation;
- activate/deactivate a single-line subscriber;
- activate/deactivate a multi-line subscriber.

B.3 *Information structure diagrams*

The information structure entities needed for the MML functions listed in § B.2 are reported in this section by means of diagrams representing each MML function information structure (Figures from B-4/Z.334 to B-23/Z.334). They are not intended to constrain in any way the enhancement of these functions in the light of technological advances or special Administrations or regulatory requirements.

The metalanguage used is described in Recommendation Z.333. In accordance with the model for single-line types (Figure B-2/Z.334) and multi-line types (Figure B-3/Z.334), the characteristics of the distinct line types can be divided into characteristics assigned lines/group of lines and characteristics assigned to subscribers. Examples for the first case are line attenuation, kind of signalling, and for the latter case abbreviated dialling, wake-up service, restrictions of the regular operation mode, etc.

In respect of these two classes of characteristics a function may require a division into two sub-functions or not, depending on system implementations and administrative needs.

Figures B-5/Z.334 to B-6/Z.334 provide an example of how this division can be accomplished, whereas the function "create a single-line subscriber" is depicted in Figure B-4/Z.334. For all the after functions this division is not covered in this annex.

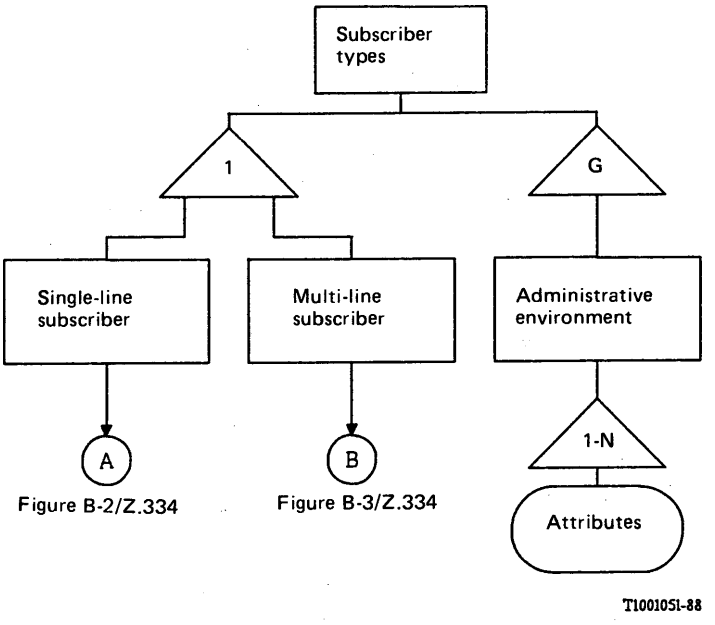


FIGURE B-1/Z.334

Subscriber types

Figure B-1/Z.334

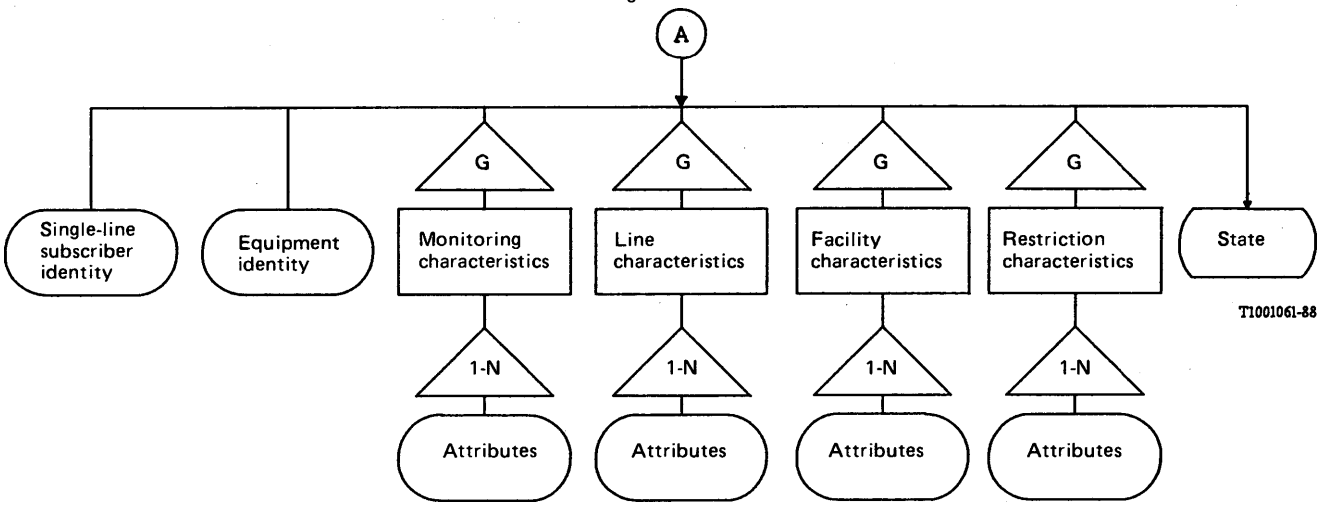


FIGURE B-2/Z.334

Model for single-line subscriber

Figure B-1/Z.334

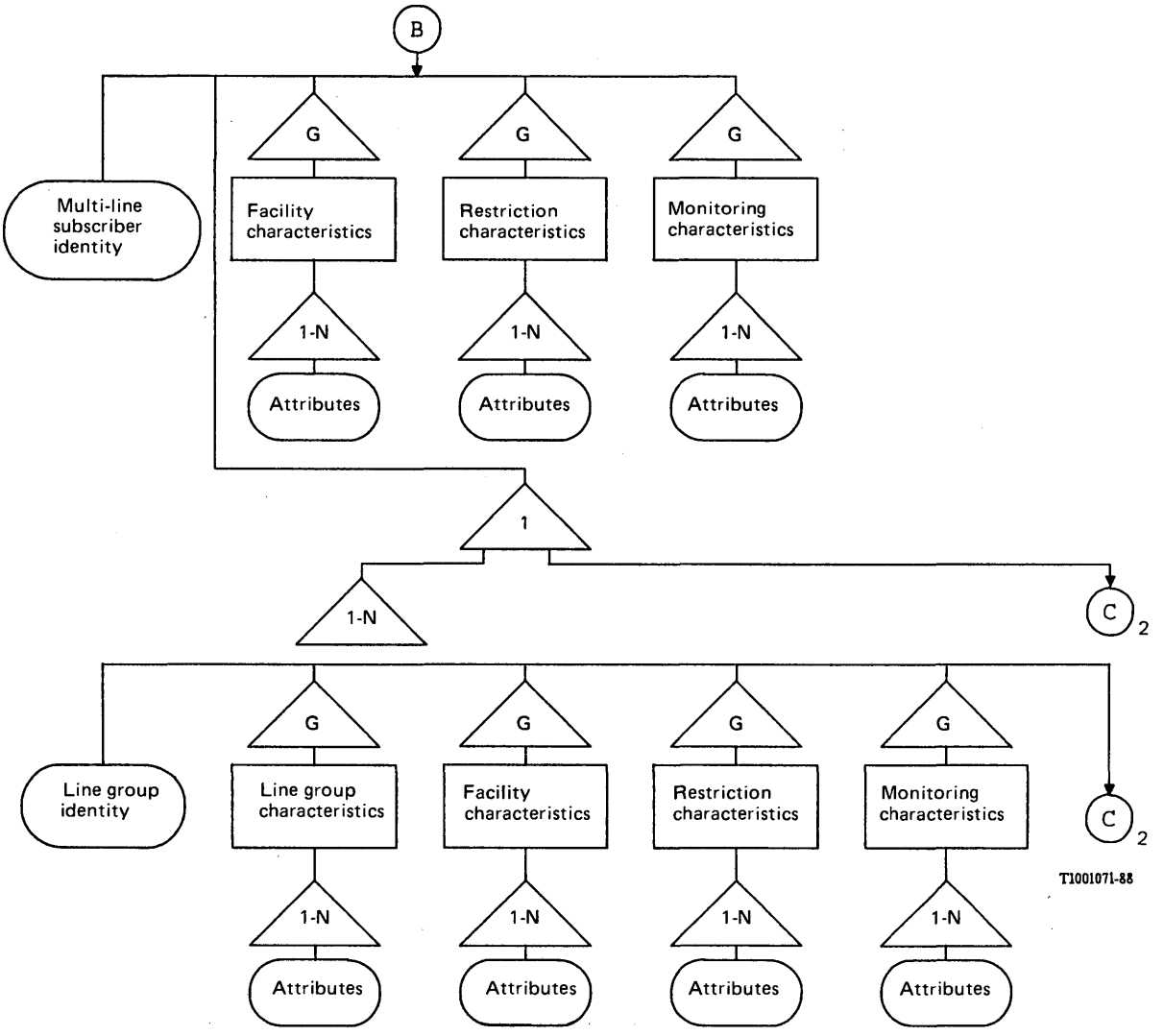


FIGURE B-3/Z.334 (1 of 2)

Model for multi-line subscriber

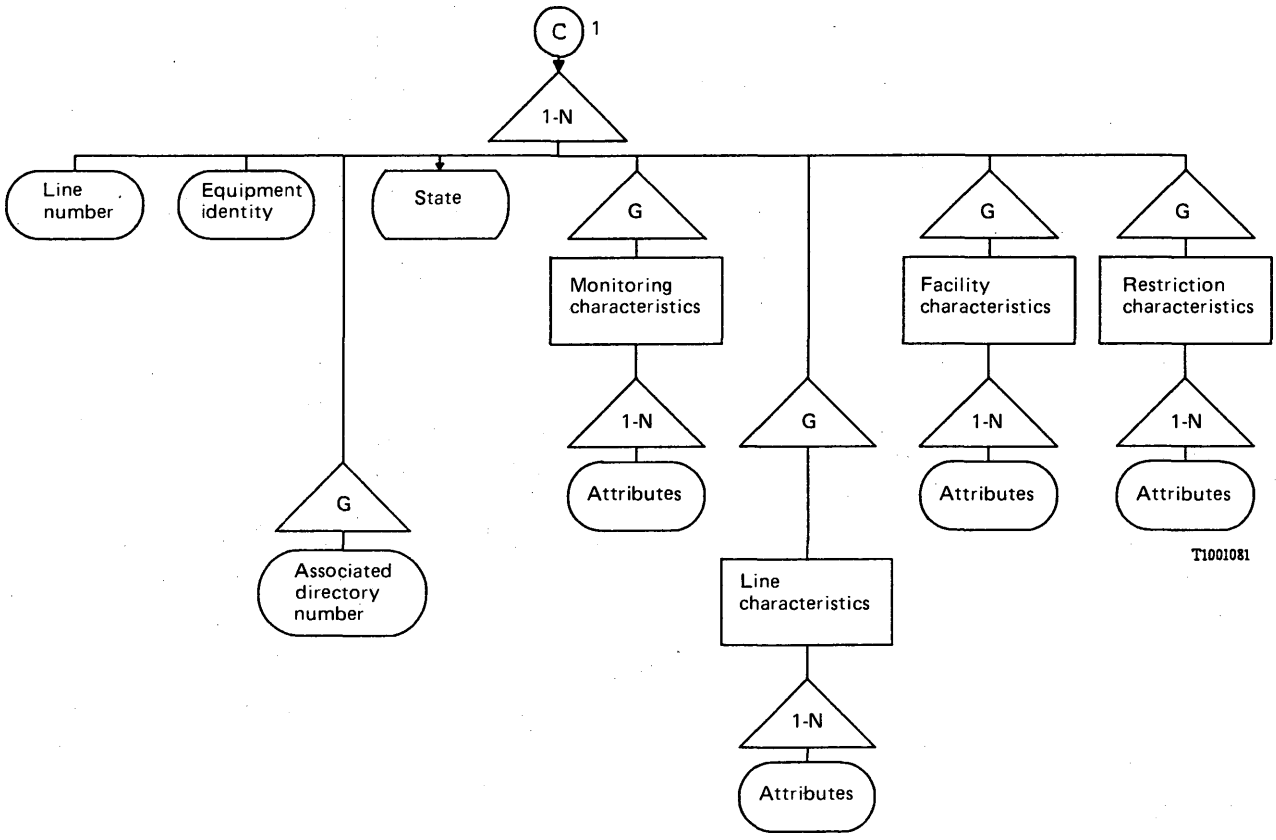


FIGURE B-3/Z.334 (2 of 2)

Model for multi-line subscriber

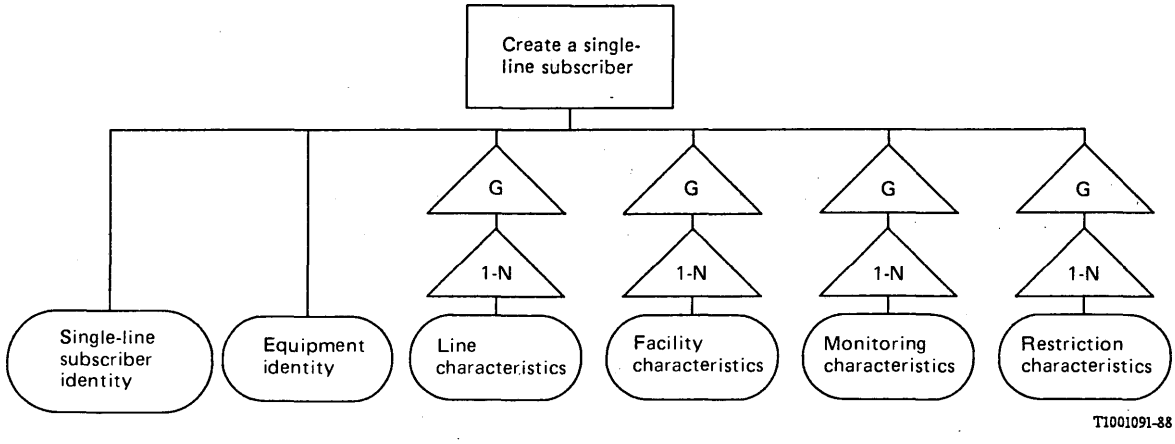


FIGURE B-4/Z.334

Create a single-line subscriber

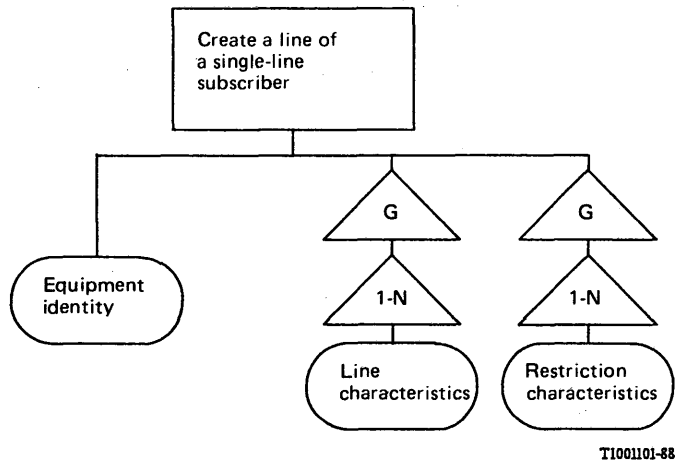


FIGURE B-5/Z.334

Create a line of a single-line subscriber

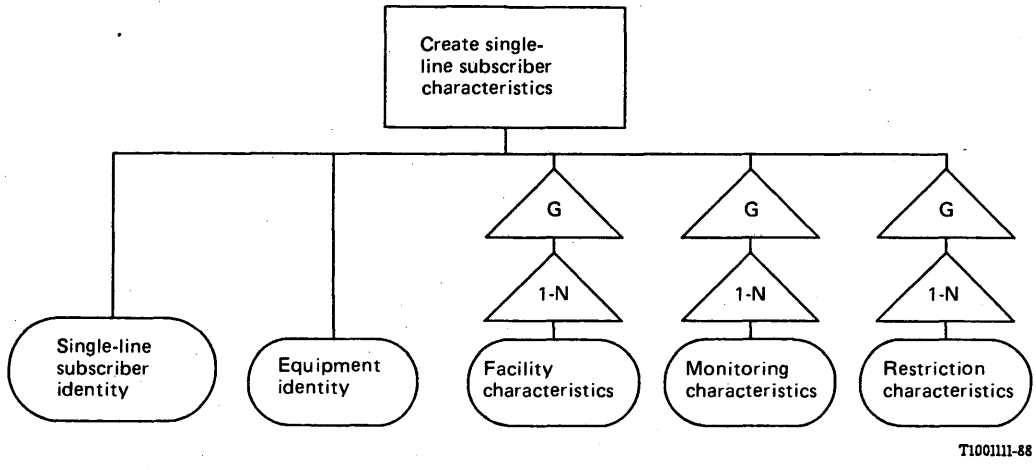


FIGURE B-6/Z.334

Create single-line subscriber characteristics

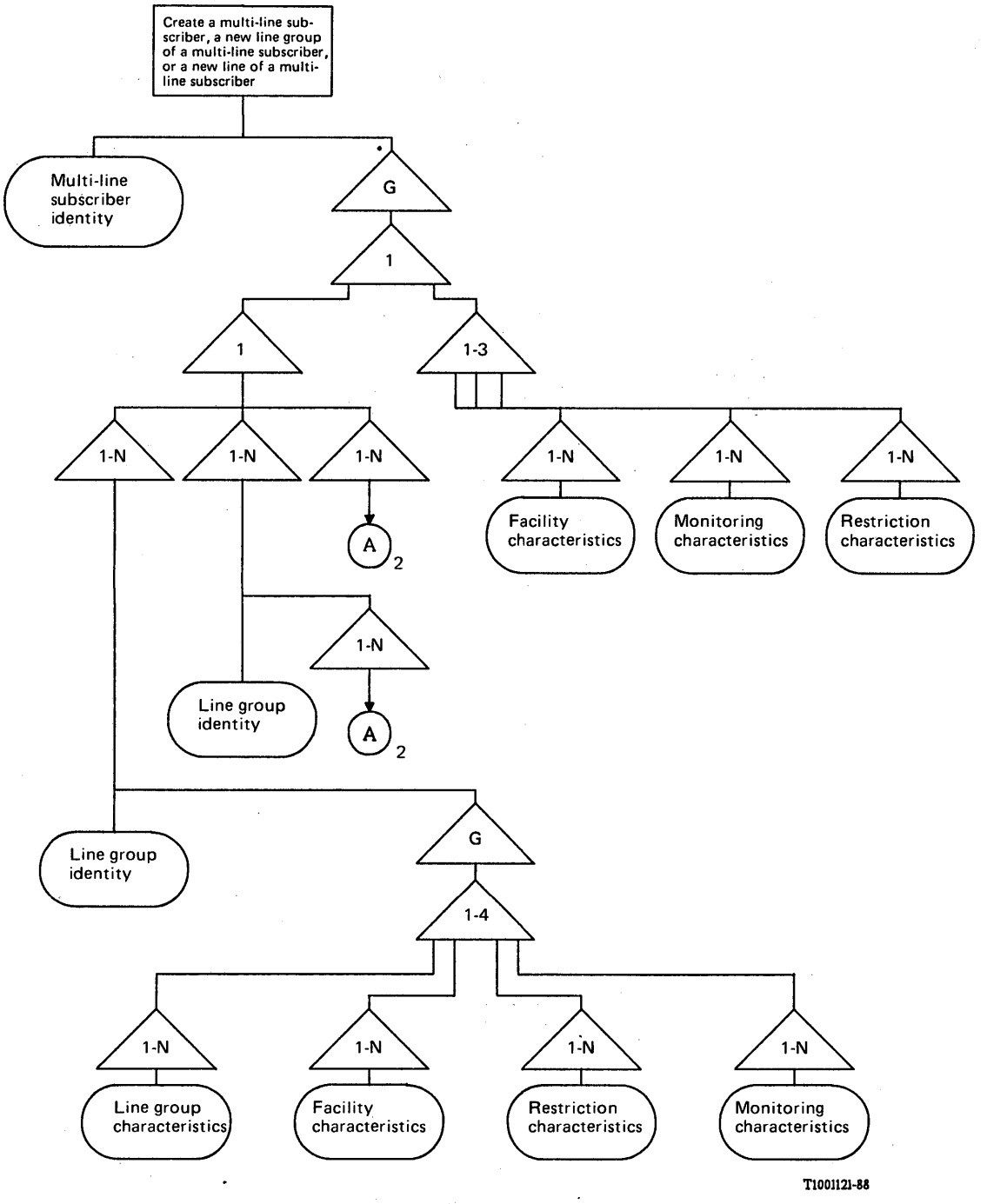


FIGURE B-7/Z.334 (1 of 2)

Create a multi-line subscriber, a new line group of a multi-line subscriber, or a new line of a multi-line subscriber

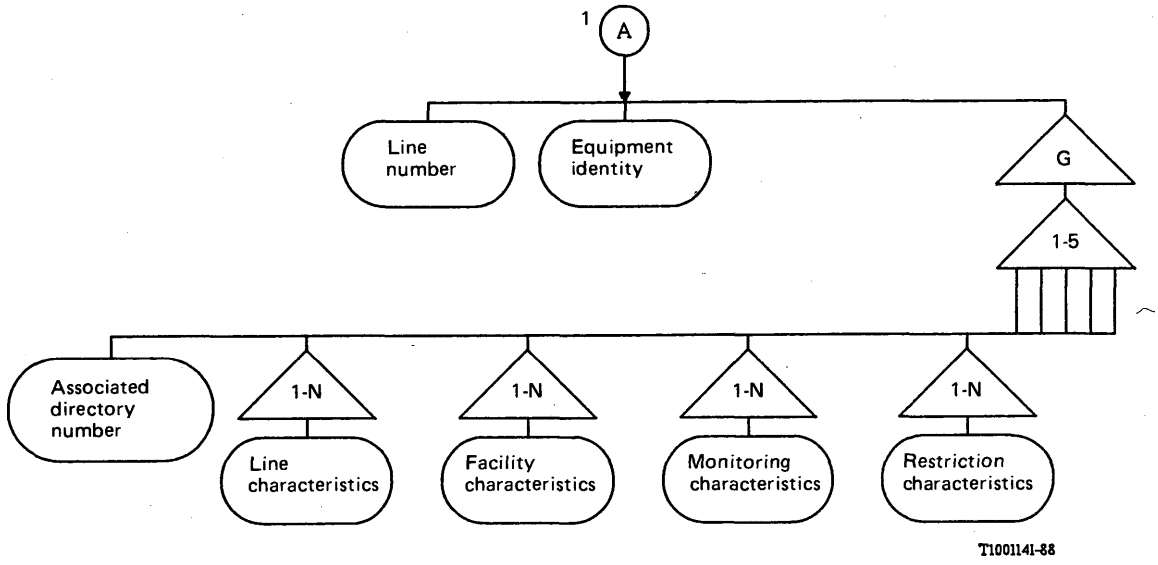


FIGURE B-7/Z.334 (2 of 2)

Create a multi-line subscriber, a new line group of a multi-line subscriber, or a new line of a multi-line subscriber

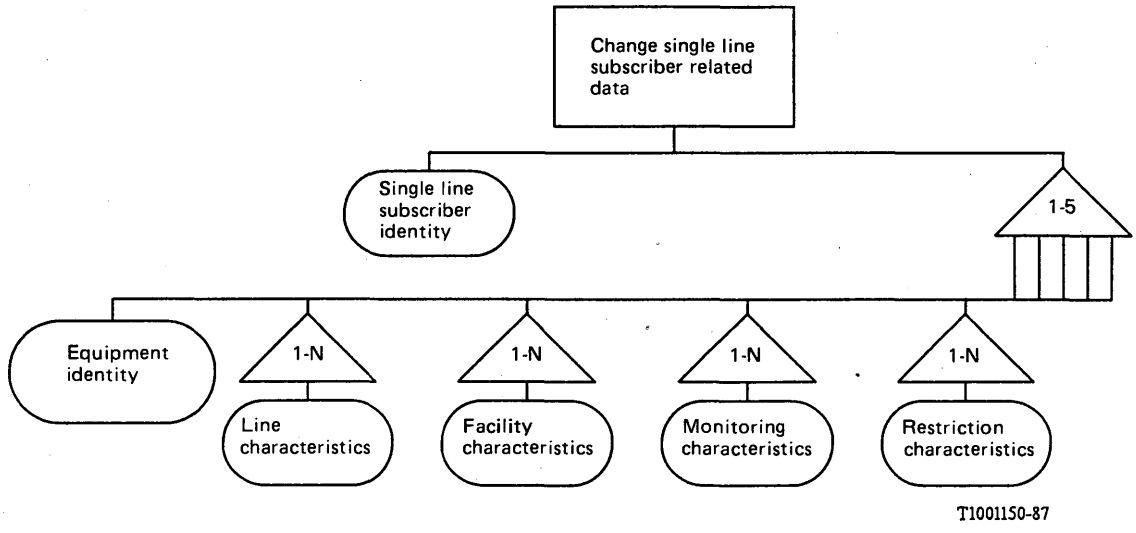


FIGURE B-8/Z.334

Change single-line subscriber related data

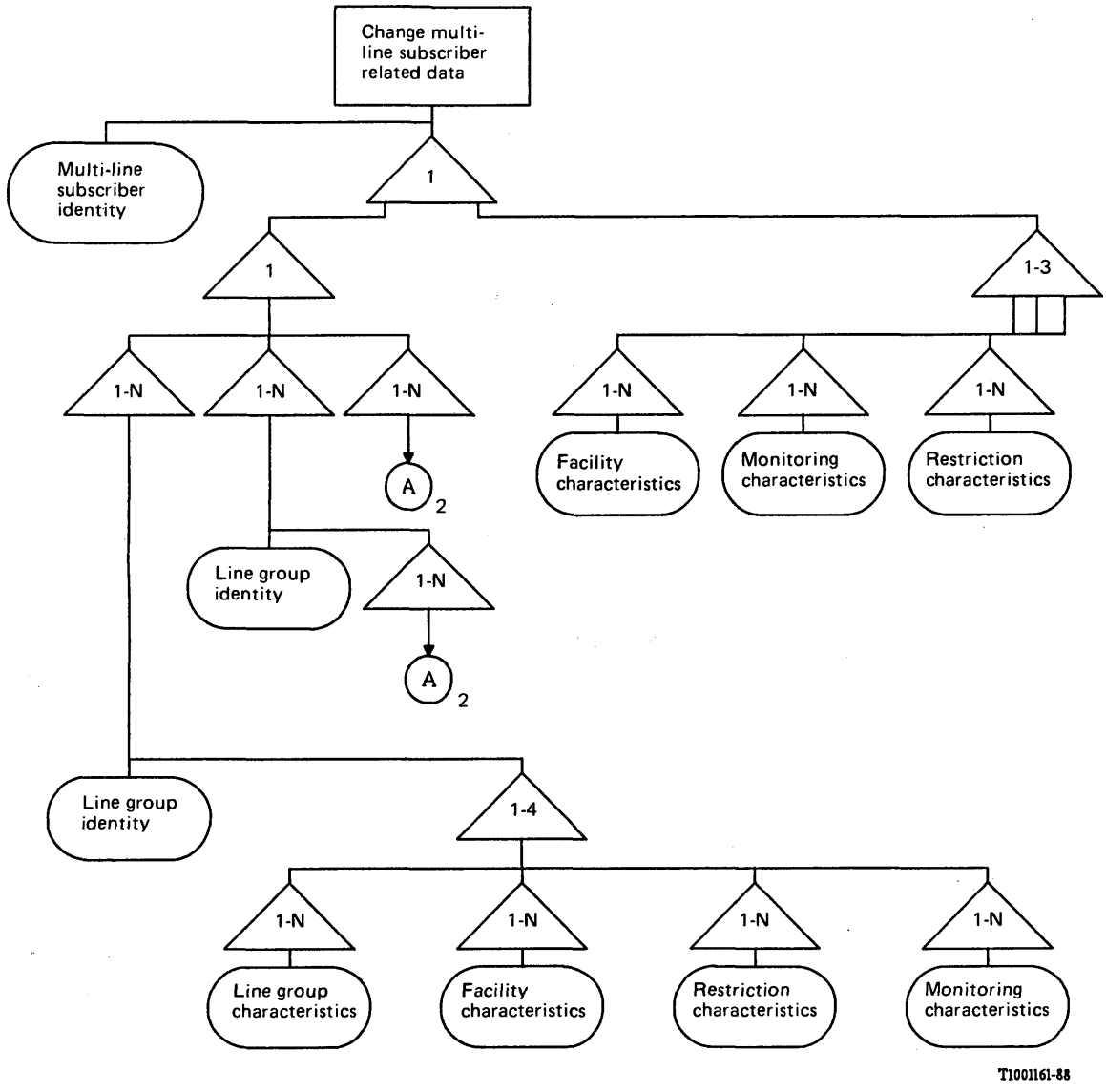


FIGURE B-9/Z.334 (1 of 2)

Change multi-line subscriber related data

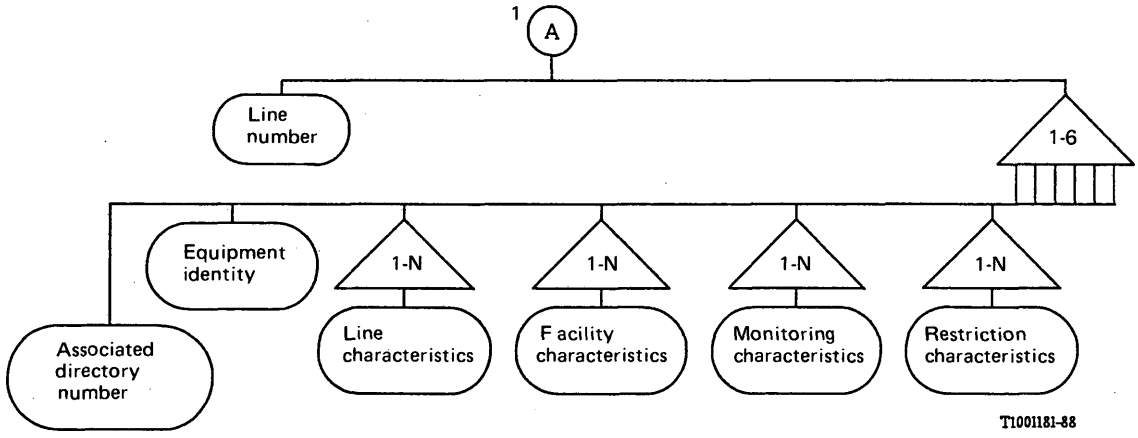


FIGURE B-9/Z.334 (2 of 2)

Change multi-line subscriber related data

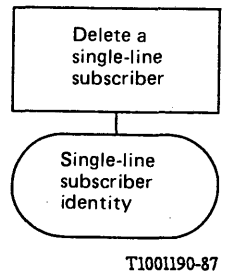


FIGURE B-10/Z.334

Delete a single-line subscriber

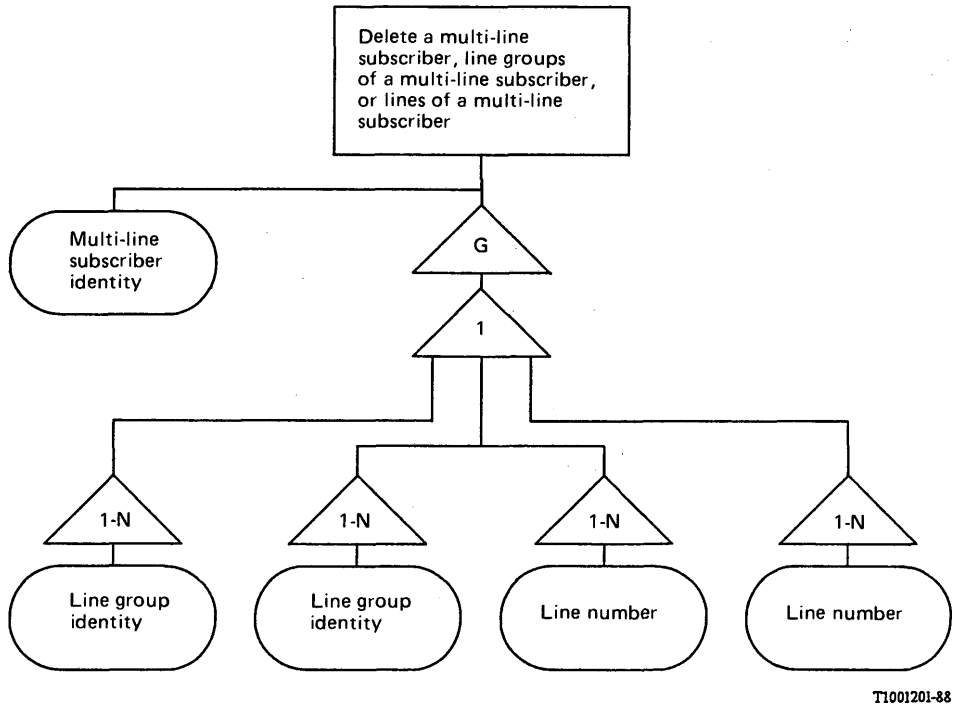
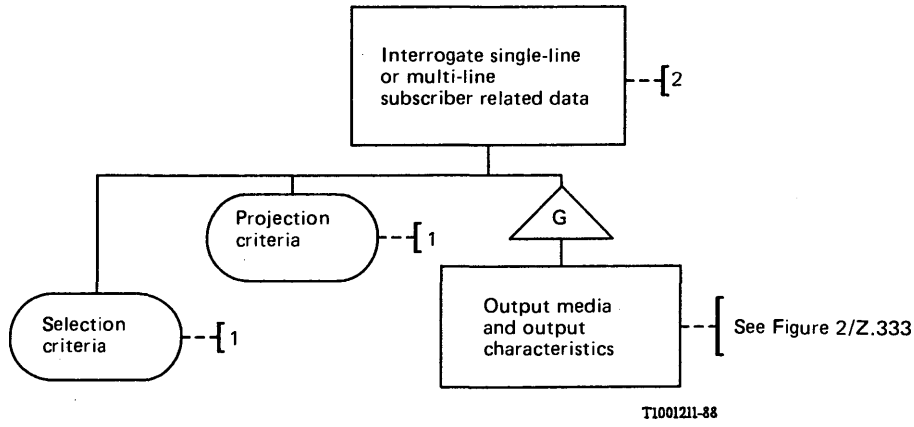


FIGURE B-11/Z.334

Delete a multi-line subscriber, line groups of a multi-line subscriber or lines of a multi-line subscriber



Note 1 – Possible selection and projection criteria:
- subscriber identity,
- equipment identity,
- single-line or multi-line subscriber identity,
- line characteristics,
- restriction characteristics,
- monitoring characteristics

Note 2 – Figures B-13/Z.334 and B-14/Z.334 contain examples of frequent interrogation functions, where the subscriber identity is the selection criterion.

FIGURE B-12/Z.334

Interrogate single-line or multi-line subscriber related data

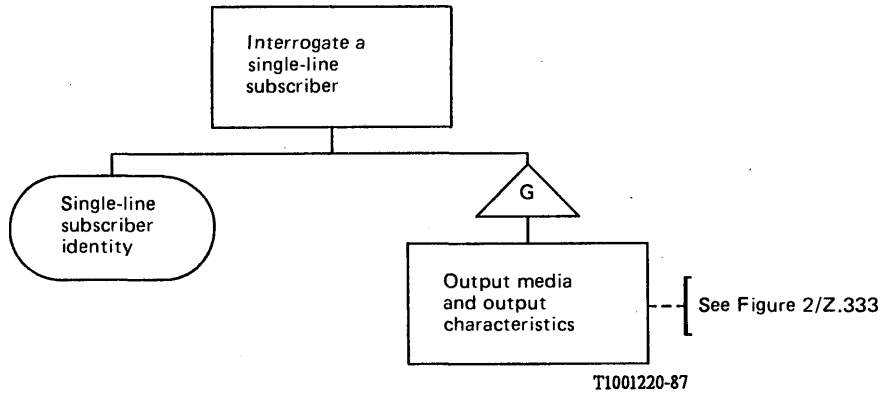


FIGURE B-13/Z.334

Interrogate a single-line subscriber

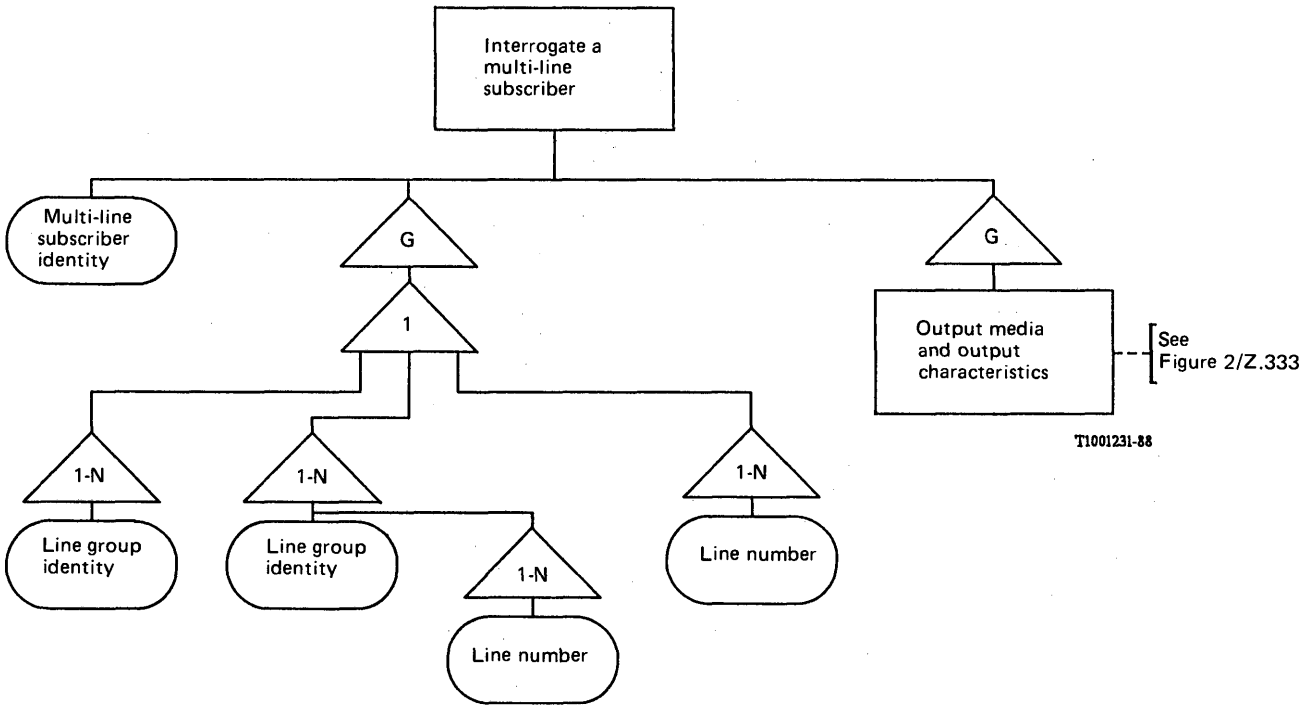


FIGURE B-14/Z.334

Interrogate a multi-line subscriber

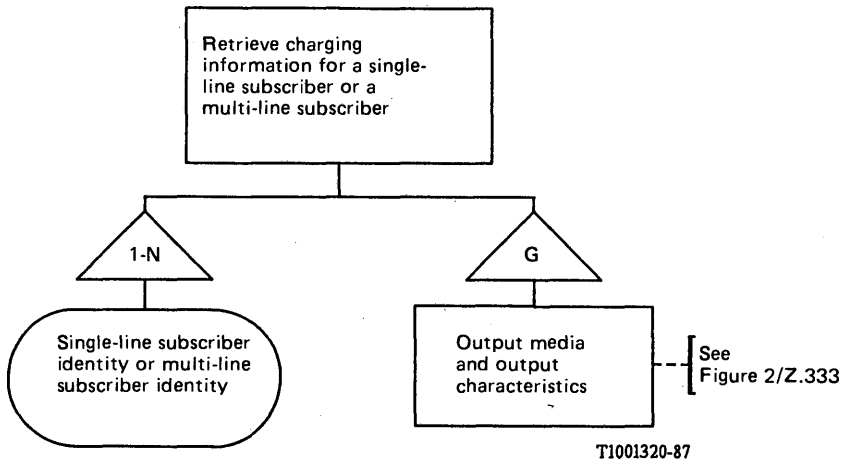


FIGURE B-15/Z.334

Retrieve charging information for single-line subscriber or multi-line subscriber

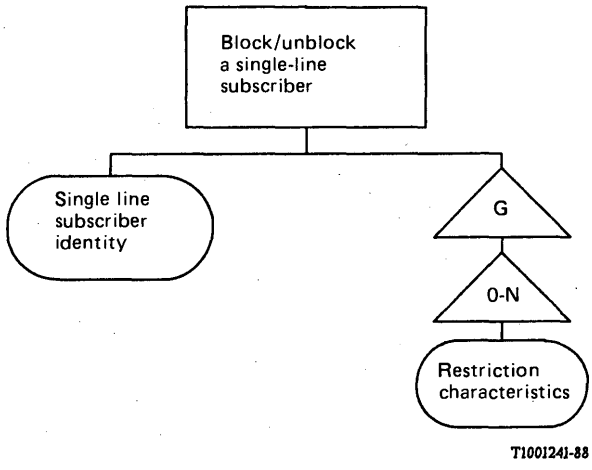
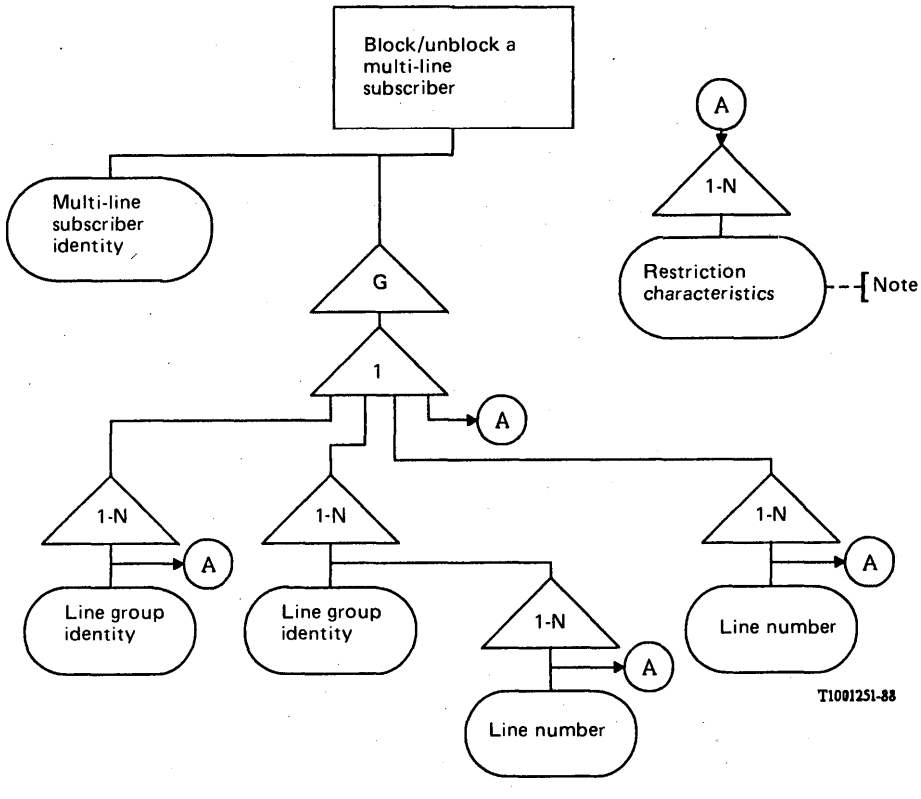


FIGURE B-16/Z.334

Block/unblock a single-line subscriber



Note – The block/unblock functions cover the following traffic restriction characteristics only:
- restricted to incoming traffic,
- restricted to outgoing traffic,
- restricted to bothway traffic.

It has been recognized, that these three traffic restrictions are not always available in existing systems.

FIGURE B-17/Z.334

Block/unblock a multi-line subscriber

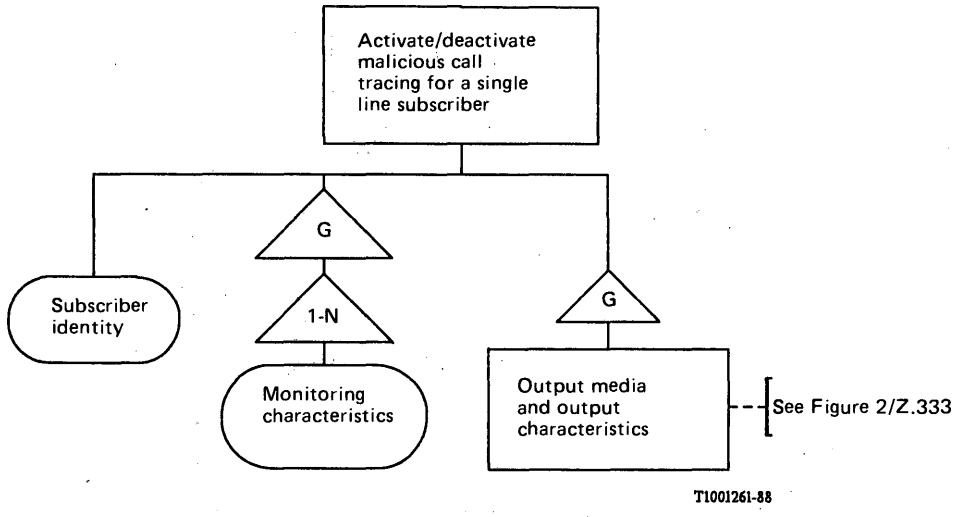
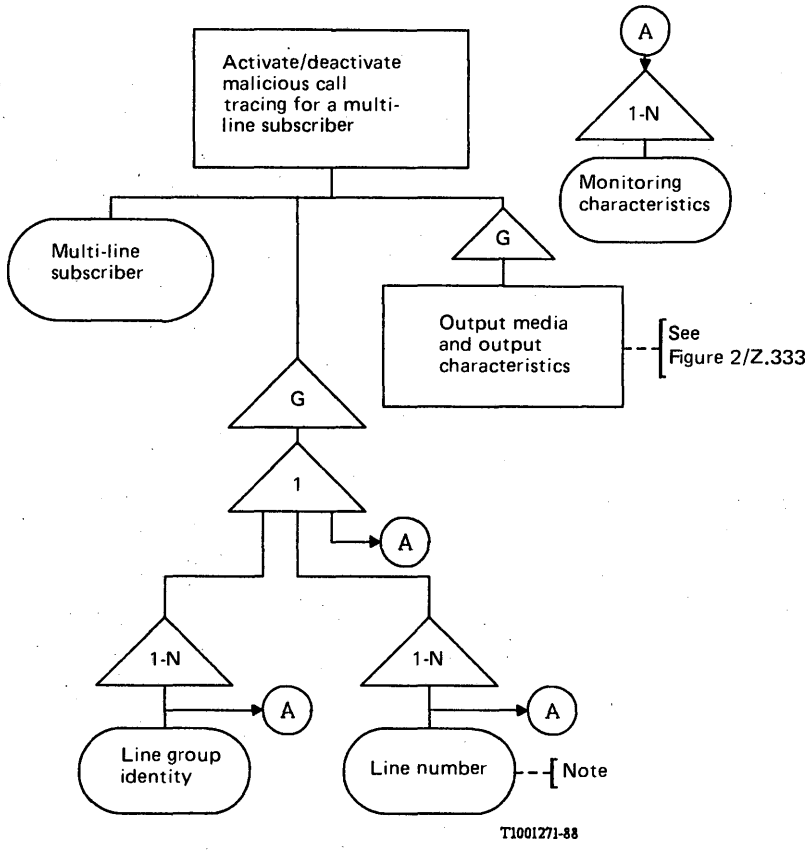


FIGURE B-18/Z.334

Activate/deactivate malicious call tracing for a single-line subscriber



Note – If the line group identity is not available, the activation/deactivation of malicious call tracing for line groups will be achieved by using the line numbers.

FIGURE B-19/Z.334

Activate/deactivate malicious call tracing for a multi-liner subscriber

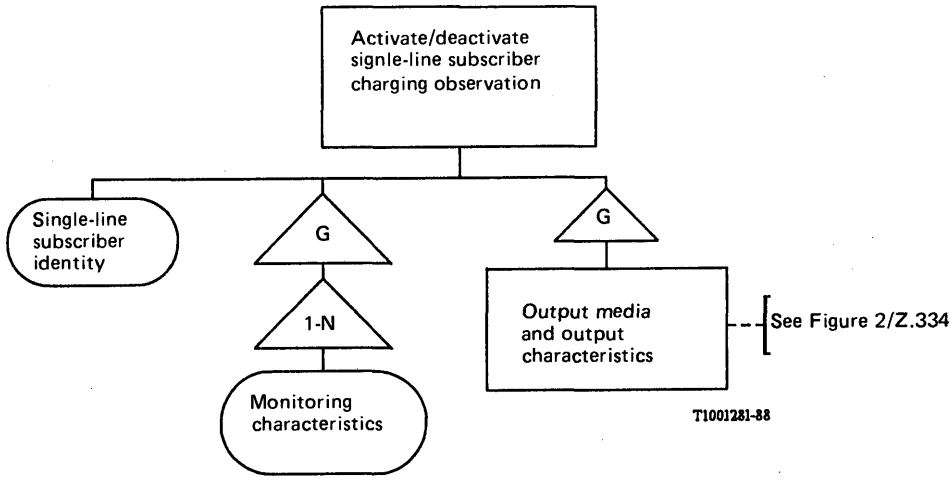


FIGURE B-20/Z.334

Activate/deactivate single-line subscriber charging observation

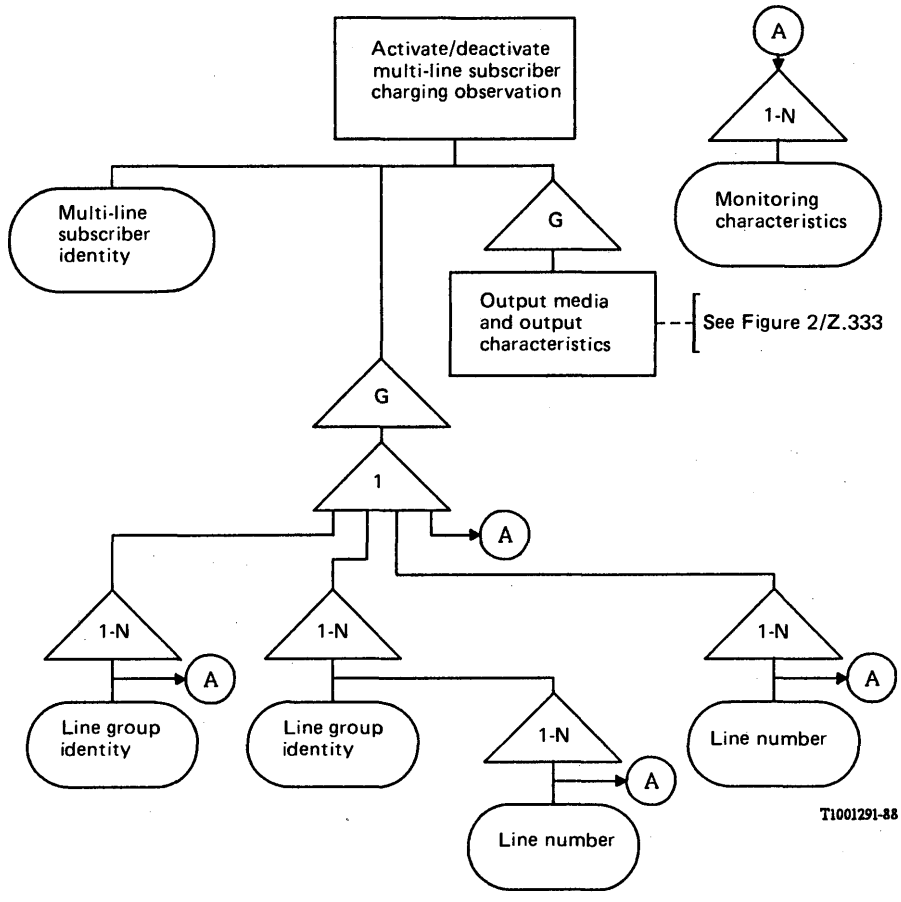
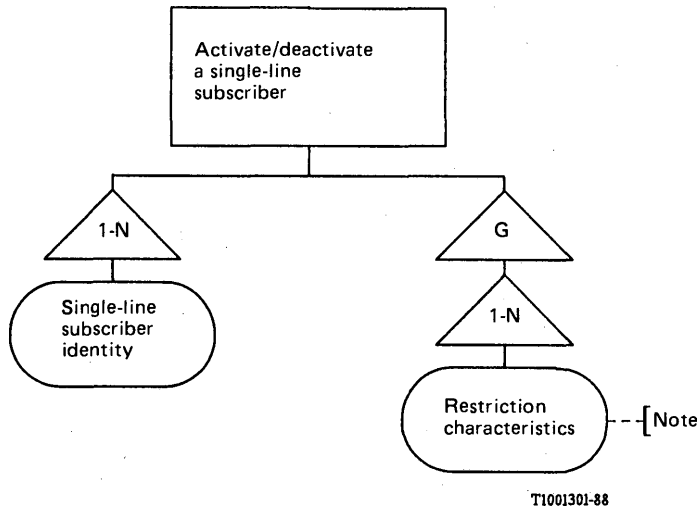


FIGURE B-21/Z.334

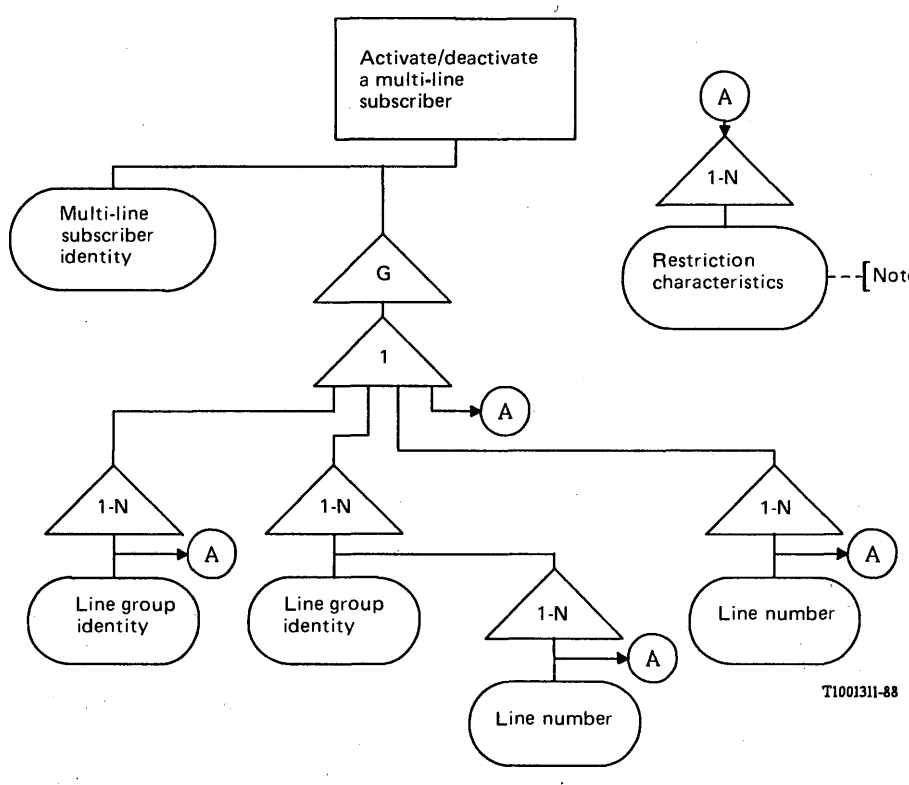
Activate/deactivate multi-line subscriber charging observation



Note – The activation/deactivation functions of a single-line subscriber manipulates the restriction characteristics in and out of service.

FIGURE B-22/Z.334

Activate/deactivate a single-line subscriber



Note – The activation/deactivation functions of a multi-line subscriber manipulates the restriction characteristics in and out of service.

FIGURE B-23/Z.334

Activate/deactivate a multi-line subscriber

Recommendation Z.335

ROUTING ADMINISTRATION

1 General

This Recommendation has been developed in accordance to the methodology defined in Recommendations Z.332 and Z.333.

The main part of this Recommendation deals with the model of routing administration and glossary of the terms used is also included.

The list of operator jobs and the list of system functions to be controlled are contained in annex A.

For each system function to be controlled by means of MML, one or more MML functions can be derived and each of them can be described using the metalanguage defined in Recommendation Z.333, in order to detail the relevant information structure.

Annex B contains a list of MML functions and information structure diagrams associated to each of them to be used as guidelines.

2 Introduction

The routing functions are those system functions that are in charge of routing a call attempt toward its destination on the basis of the data associated with the call attempt (e.g. the dialled digits, etc.) and the data associated to the network (e.g. identities of the circuit sub-groups serving a certain destination, etc.). The destination of a call attempt may be inside the switching system as well as outside the switching system. If the destination is inside the switching systems, the routing will consist of identifying the termination that corresponds to the destination. If the destination is outside the switching system the routing functions shall search for a free circuit, within a given circuit sub-group on which the call attempt will be routed.

Routing consists of functions whose parameters may vary with the network design and the switching system design to fit the network application. The economics of a switching system in a network application is, in part, a function of how well the routing functions match the network design.

The main activities for creating and maintaining routing administration data may be summarized by the following items:

- analysis of the data provided by the traffic measurement functions;
- defining the routing strategies on the basis of the achieved results;
- creating new or changing the existing items, related to the routing in the switching systems, in order to satisfy the routing strategies.

Only the last of these items is within the scope of this Recommendation.

This Recommendation deals with the routing administration functions involved with the routing towards destinations outside the switching system, whichever signalling system is used.

3 Routing administration model

3.1 *Introduction*

To service a call in order to make it proceed towards the desired called party, two steps are pointed out:

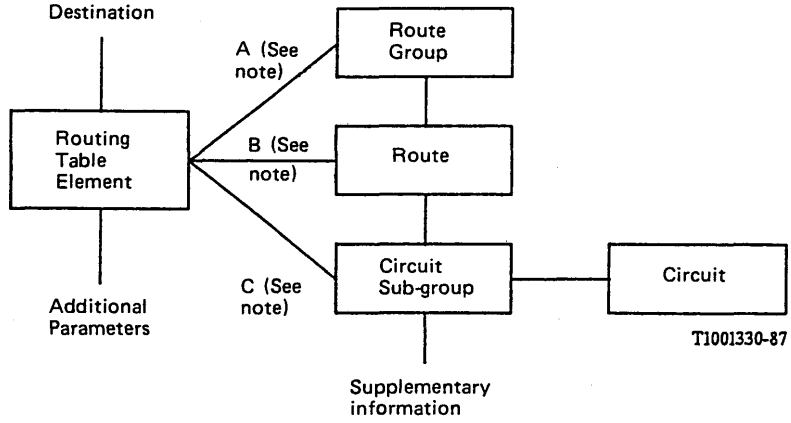
- the identification of the destination of the call made by means of the analysis of the selection digits and by means of other information;
- the selection of a free circuit within a set of suitable circuits on which the call may be forwarded.

The routing functions provided by a system deal mainly with the second step and the present Recommendation relates to their administration. In this section a model for the administration of routing functions is reported, describing the information used.

Figure 1/Z.335 provides a description of the connections that exist at the man-machine interface, among the routing items (such as circuits, circuit sub-groups, routes and route groups) that can be administered by means of MML functions.

The purpose of this model is to provide an overview of the routing items and the relevant parameters. No specific association between parameters and routing items is intended other than for explanatory purposes. Any parameter can be associated to any routing item depending on the specific needs of an administration or of the network: for example, the type of signalling system may be determined at route group, route or circuit sub-group level, without changing its function.

The collection of combinations of destination and additional parameters forms the routing table, which contains all the possibilities of routing for a given system. One entry in this table is called routing table element.



Note – Option A = The model comprises route groups, routes and circuit sub-groups.
Option B = The model comprises routes and circuit sub-groups.
Option C = The model comprises circuit sub-groups only.
These options are to be considered, on an equal footing, as complete models of equal acceptability. Even though route groups and routes can be implicitly present in the routing table, they do not necessarily appear explicitly in the man-machine interface.

FIGURE 1/Z.335

Routing administration model

3.2 *Parameters*

The parameters used in the decision process of routing a call are included in the following list. In Figure 1/Z.335 they are represented by destination, additional parameters and supplementary information. The list is open-ended and not all the parameters are needed for all applications. The list may vary according to administration needs, telecomunication network levels, etc.

The methods used to evaluate these parameters within the switching system should not be constrained by the model.

1) Destination;
2) Class of incoming circuit sub-group;
3) Calling party category;
4) Parameters for network management (e.g. time of day, presence of carrier failure, etc.);
5) Signalling system (incoming and outgoing);
6) Transmission characteristics (incoming and outgoing);

7) Circuit type (e.g. two wires, four wires);

8) Presence of echo canceller;

9) Transmission medium characteristics (i.e. presence of satellite links);

10) Selection of specific carrier or specific network.

The destination is identified by means of the selection digits on the basis of the incoming route (origin information) or signalled information.

The additional parameters are those parameters that allow identification of the possible choices for routing a call towards a given destination. The following factors may be used as additional parameters (the list is neither mandatory nor complete):

a) Class of incoming circuit sub-group;

b) Calling party category;

c) Parameters for network management.

A combination of destination and additional parameters identify a routing table element which contains all the possibilities to set up a call in order to forward it.

3.3 *Route group*

A route group consists of one or more routes that may be used for a given destination. A route group may be accessed by more than one combination of destination and additional parameters.

3.4 *Route*

A route consists of one or more circuit sub-groups and of some information associated with the route describing the use of the route itself.

Examples of this information may be:

- seizure signal to send (terminal, transit);
- number of digits to send;
- digits to add.

If the route is not present in the routing administration model the information should be associated to other routing items.

3.5 *Circuit sub-group*

A circuit sub-group is a set of circuits between two exchanges. In options A and B, circuits within a circuit sub-group have similar characteristics (e.g. type of signalling, type of transmission path, etc.).

The selection of a given circuit sub-group inside a route can be made on the basis of supplementary information such as signalling system, transmission technique, circuit type (e.g. two wires, four wires), presence of echo canceller, transmission medium characteristics (e.g. presence of satellite links), etc. Other information may also be associated to circuit sub-groups; examples of this information are:

- traffic direction;
- class of incoming circuit sub-group;
- search method to select a circuit inside the sub-group.

3.6 *Circuit*

A circuit is characterized by an identity, by its hardware termination identity and by its sequence number inside the circuit sub-group (this latter is only used for selection purposes in outgoing circuit sub-groups). If common channel signalling is used, a label should be assigned to the circuit.

4 Additional information

4.1 *Management of the status of routing items*

In order to change the routing capabilities of the telecommunication systems, facilities are required to alter the operating status of the items involved (i.e. to put a circuit in service or out of service, etc.).

This topic is left for further study.

4.2 *Reliability mechanisms*

In order to prevent malfunctions and errors when servicing a call, the routing administration functions used in the telecommunication system must implement reliability mechanisms. The most critical functions, e.g. change functions, should not be provided to the operator if reliability mechanisms are not provided.

The selection and implementation of reliability mechanisms are not covered in this Recommendation.

However, due to the fact that the machanism used may affect the man-machine interface, it is highlighted that such aspect should be studied and taken into account when designing the real man-machine interface.

5 Glossary of used terms

circuit

Connection between two exchanges for one call at a time, including the junctors that terminate the circuit.

circuit sub-group

Group of circuits between two exchanges having the same traffic direction (incoming, outgoing, bidirectional), the same signalling characteristics and the same transmission medium characteristics.

route

Collection of circuit sub-groups between two exchanges that are equivalent for routing purposes. The term route is equivalent to the concept of "circuit group" as used in Recommendation Z.337 and in E-Series Recommendations.

route group

The set of all the possible routes on which a call may be forwarded to the appropriate destination.

ANNEX A

(to Recommendation Z.335)

List of system functions to be controlled by MML and list of jobs

A.1 *List of system functions to be controlled by MML*

1) Managing the routing data base
2) Querying the routing data base

A.2 *List of jobs*

1) To create/change items involved in the routing functions or change old ones (e.g. circuit sub-groups, routes, etc.)
   - the purpose of the job is to create and/or change the configuration of the switching system regarding the items involved in the routing of a call;
   - the system is supposed to record the new data and to check their static correctness;

- the operator is supposed to input all needed data;
- the complexity of the job may be high depending on the amount of the data to be input for the item to be created/changed;
- the frequency of the job is medium;
- the job is supposed to be performed at exchange and/or OMC level.

2) To delete items related to the routing
- the purpose of the job is to delete items no longer useful;
- the system is supposed to delete the specified items deleting the related data only if no call is currently routed on the item;
- the operator is supposed to introduce the identity of the item to be deleted;
- the complexity of the job is low (if the system makes sufficient checks);
- the frequency of the job is low;
- the job is supposed to be performed at exchange and/or OMC level.

3) To interrogate different kinds of information related to the routing
- the purpose of the job is to get information on the current data used by the system to perform the routing;
- the system is supposed to output the information requested in suitable formats and on the selected devices;
- the operator is supposed to input the identity of the item to be interrogated and to select the information to be retrieved;
- the complexity of the job is low;
- the frequency of the job is high;
- the job is supposed to be performed at exchange and/or OMC level.

ANNEX B

(to Recommendation Z.335)

Guidelines for the list of MML functions and associated information structure diagrams

B.1 *Introduction*

This annex contains guidelines for the list of MML functions and associated information structure diagrams related to the routing administration model defined in Recommendation Z.335, § 3.

B.2 *List of MML functions*

This list contains possible MML functions for routing administration.

This list is not mandatory nor complete, it may vary according to administration needs, telecommunication network levels, regulatory needs, etc.

These MML functions do not represent the actual command structure of any real implementation of the man-machine interface. Each of the MML functions identified can be implemented by providing one or more separate distinctive commands or several MML functions could be implemented by using a single command.

1) *Creation*
- create a circuit,
- create a circuit sub-group,
- create a route (only options A and B)[1],
- create a route group (only option A)[1],
- create a destination.

2) *Deletion*
- delete a circuit,
- delete a circuit sub-group,

[1] Options A and B are defined in Recommendation Z.335, § 3.

- delete a route (only options A and B)[2],
- delete a route group (only option A)[2],
- delete a destination.

3) *Interrogation*
   - interrogate a circuit,
   - interrogate a circuit sub-group,
   - interrogate a route (only options A and B)[2],
   - interrogate a route group (only option A)[2],
   - interrogate a destination.

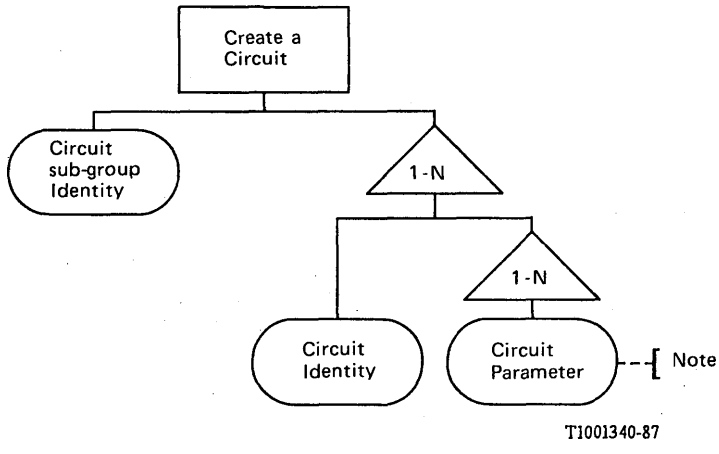
4) *Changing*
   - change a circuit,
   - change a circuit sub-group,
   - change a route (only options A and B)[2],
   - change a route group (only option A)[2],
   - change a destination.

B.3 *Information structure diagrams*

Only the information entities needed for the MML functions previously derived have been identified. They are reported in this section by means of diagrams representing each MML function information structure.

These diagrams represent typical routing function requirements. They are not intended to constrain in any way the enhancement of these functions in the light of technological advances or specific administration or regulatory requirements.

The metalanguage used is described in Recommendation Z.333.

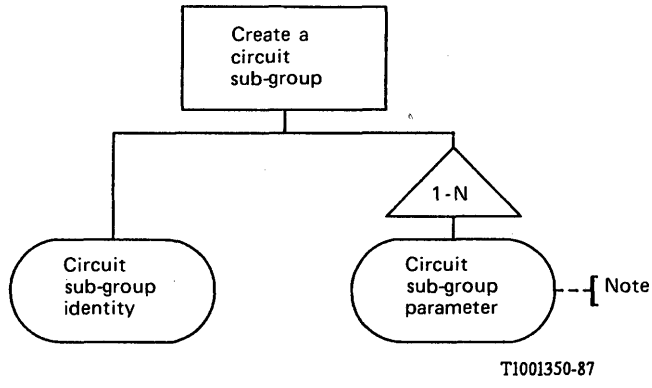


Note – Possible parameters are:
- HW terminal identity.
- Sequence number.
- Common channel signalling label.

FIGURE B-1/Z.335

Create a circuit

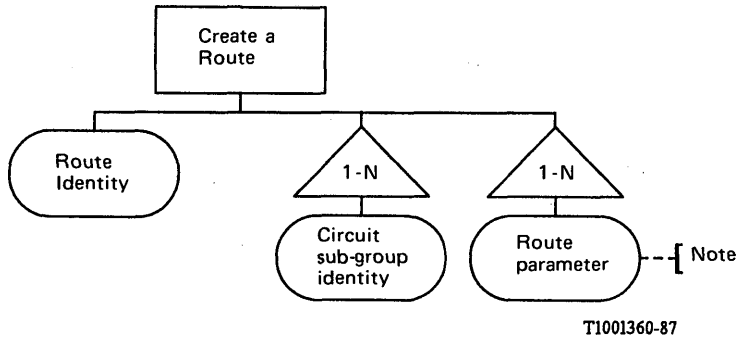
[2] Options A and B are defined in Recommendation Z.335, § 3.



Note – Possible parameters are:
- Signalling system.
- Traffic direction.
- Circuit type.
- Transmission technique.
- Presence of echo canceller.
- Transmission medium characteristics.
- Class of incoming circuit sub-group.

FIGURE B-2/Z.335

Create a circuit sub-group



Note – Possible parameters are:
- Type of seizure to send.
- Digits to be added.
- Number of digits to suppress.
- Search method.

FIGURE B-3/Z.335

Create a route

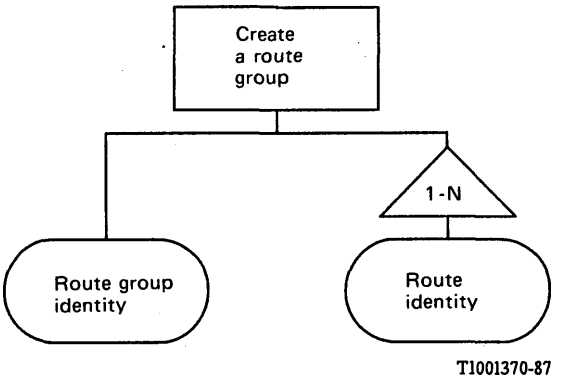
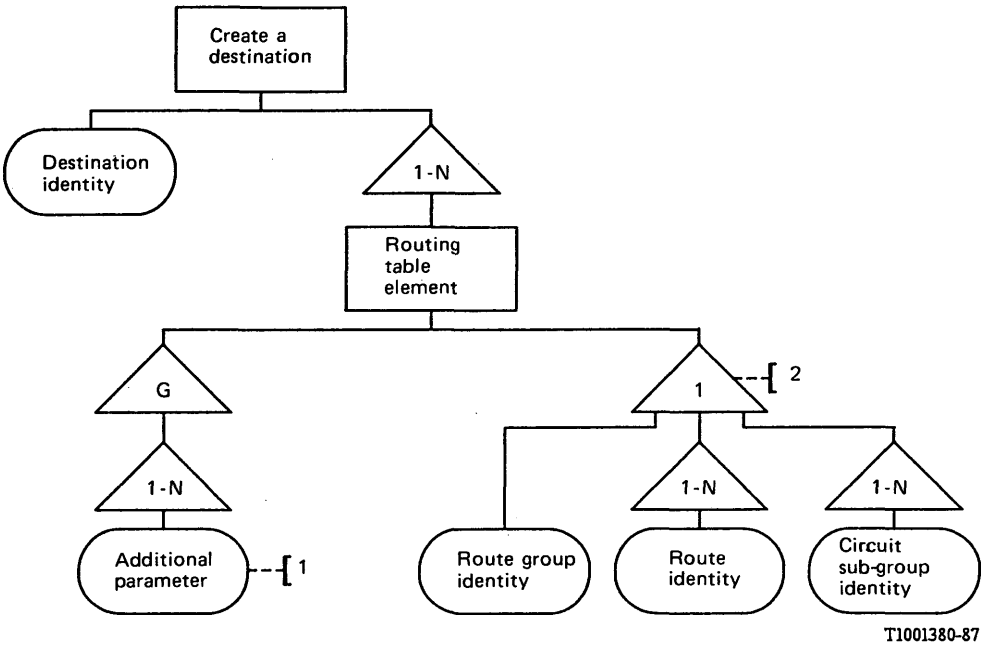


FIGURE B-4/Z.335

Create a route group



Note 1 – Possible parameters are:
- Calling party category.
- Class of incoming circuit group.
- Parameters for network management.

Note 2 – The selection is made according to the option (A, B or C) adopted.

FIGURE B-5/Z.335

Create a destination

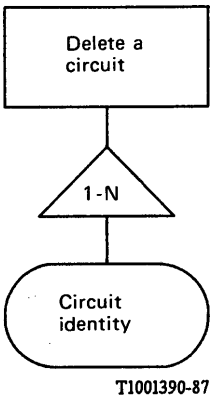


FIGURE B-6/Z.335

Delete a circuit

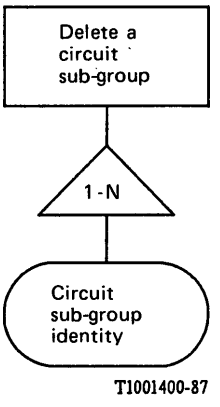
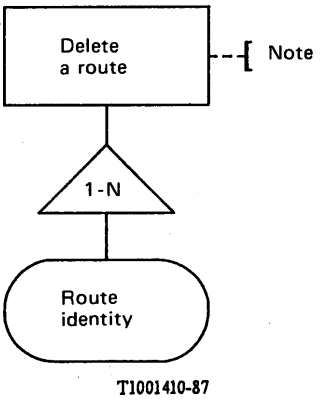


FIGURE B-7/Z.335

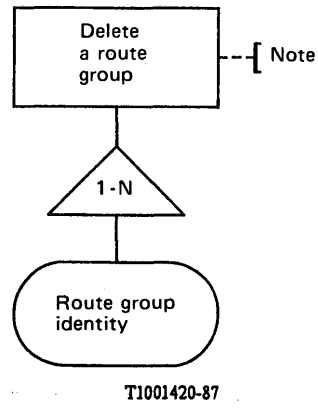
Delete a circuit sub-group



Note – Only required in option A and B.

FIGURE B-8/Z.335

Delete a route



Note – Only required in option A.

FIGURE B-9/Z.335

Delete a route group

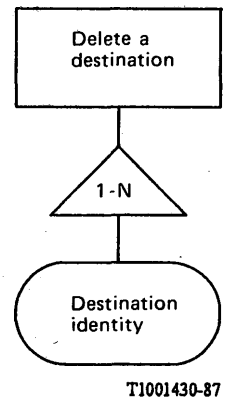
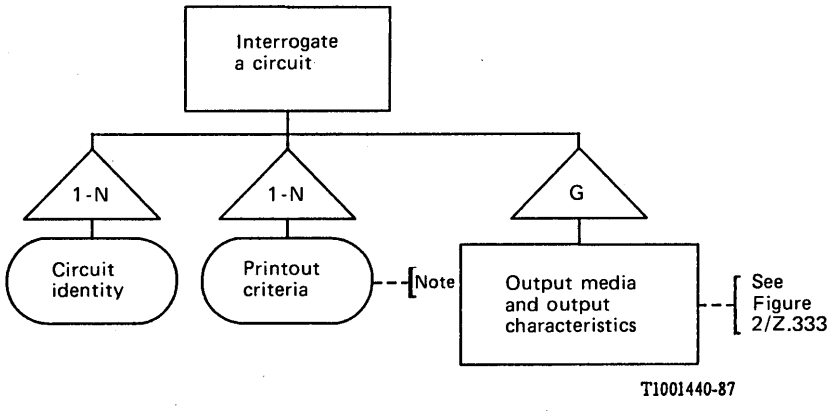


FIGURE B-10/Z.335

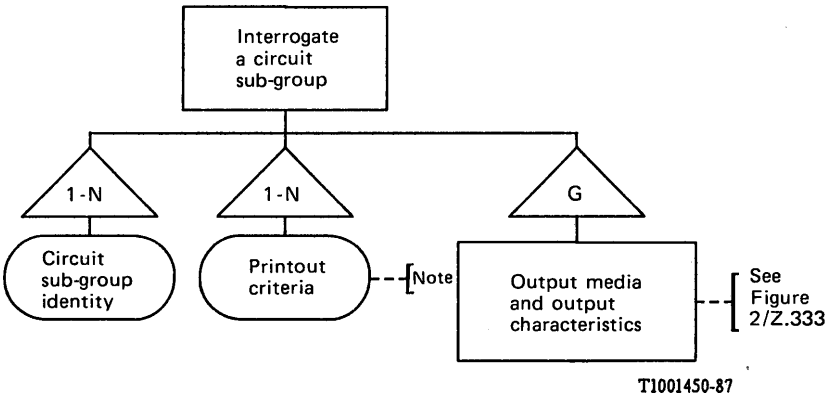
Delete a destination



Note – Possible values:
- Status.
- HW termination identity.
- Sequence number.
- Circuit sub-group owning the circuit.

FIGURE B-11/Z.335

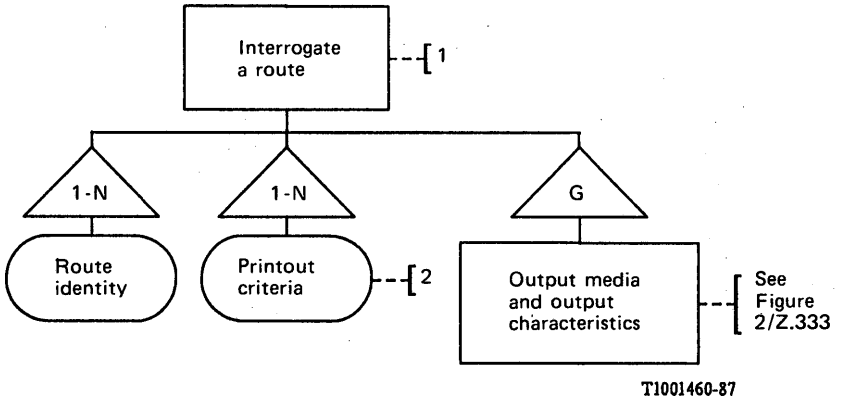
Interrogate a circuit



Note – Possible values:
- Circuit identities.
- Signalling system.
- Traffic direction.
- Transmission medium characteristics.
- Transmission technique.
- Presence of echo canceller.
- Circuit type.
- Search method.
- Routes owning the circuit sub-group.

FIGURE B-12/Z.335

Interrogate a circuit sub-group

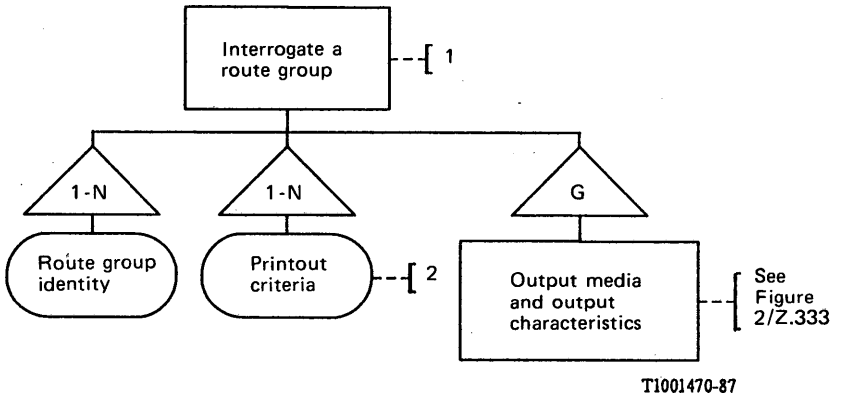


Note 1 – Only required in options A and B.

Note 2 – Possible values:
- Circuit sub-groups.
- Circuit sub-groups sequence.
- Traffic direction.
- Route group owning the route (only option A).
- Type of seizure and digits to add (only incoming).
- Type of seizure, digits to add or to suppress (only option B).
- Destinations vs additional parameters associated to route (only option B).

FIGURE B-13/Z.335

Interrogate a route

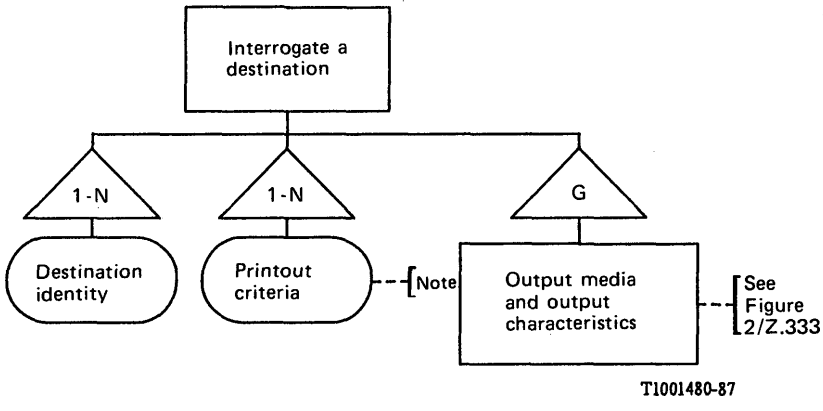


Note 1 – Only required in option A.

Note 2 – Possible values:
- Route identities.
- Destinations owning the route group.
- Destinations vs additional parameters.

FIGURE B-14/Z.335

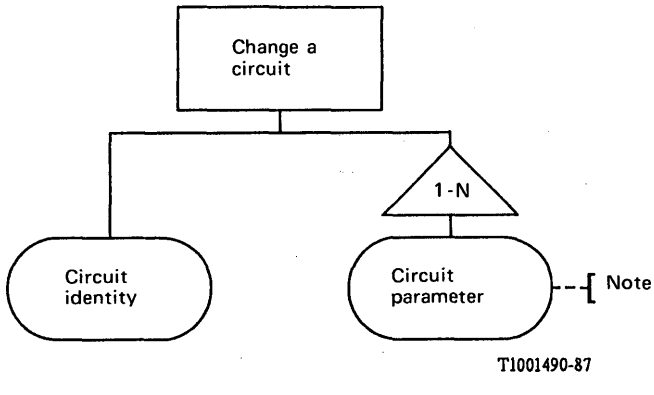
Interrogate a route group



Note – Possible values:
- Route groups (option A).
- Route groups vs additional parameters (option A).
- Routes (option B).
- Routes vs additional parameters (option B).
- Circuit sub-groups (option C).
- Circuit sub-groups vs additional parameters (option C).

FIGURE B-15/Z.335

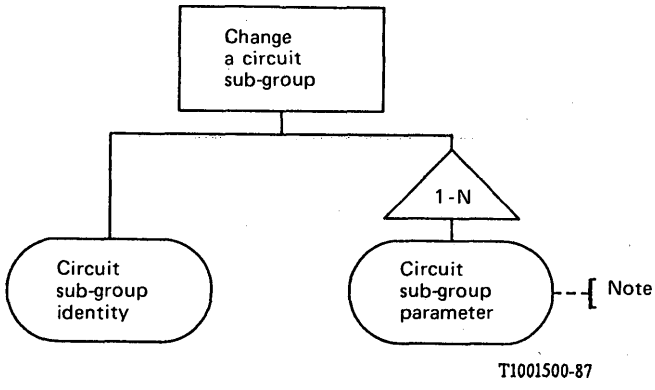
Interrogate a destination



Note – Possible parameters are:
- HW terminal identity.
- Sequence number.
- Common channel signalling label.

FIGURE B-16/Z.335

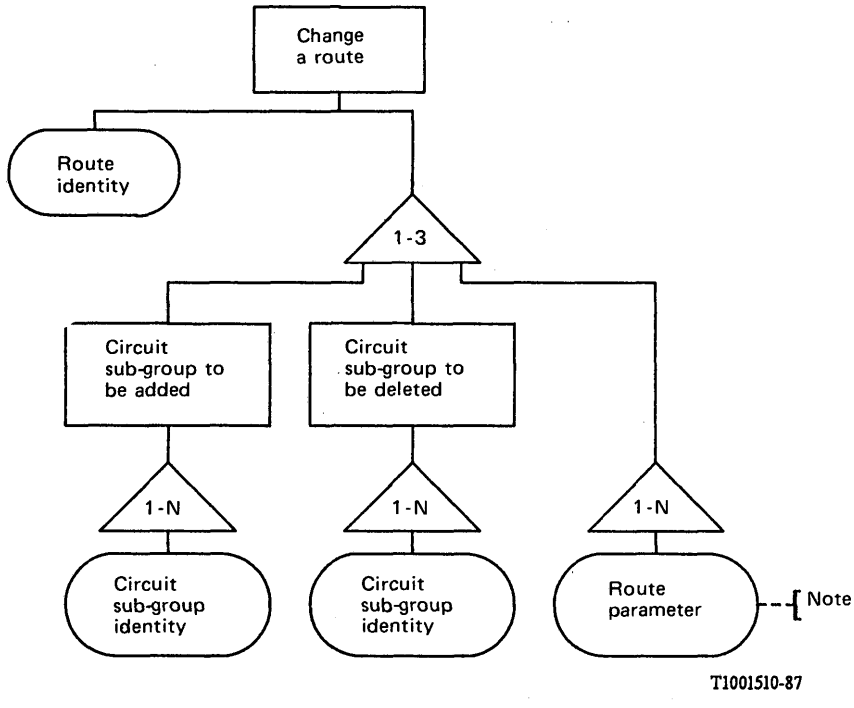
Change a circuit



Note – Possible parameters are:
- Signalling system.
- Traffic direction.
- Circuit type.
- Transmission technique.
- Presence of echo canceller.
- Transmission medium characteristics.
- Class of incoming circuit sub-group.

FIGURE B-17/Z.335

Change a circuit group



Note – Possible parameters are:
- Type of seizure to send.
- Digits to be added.
- Number of digits to suppress.
- Search method.

FIGURE B-18/Z.335

Change a route

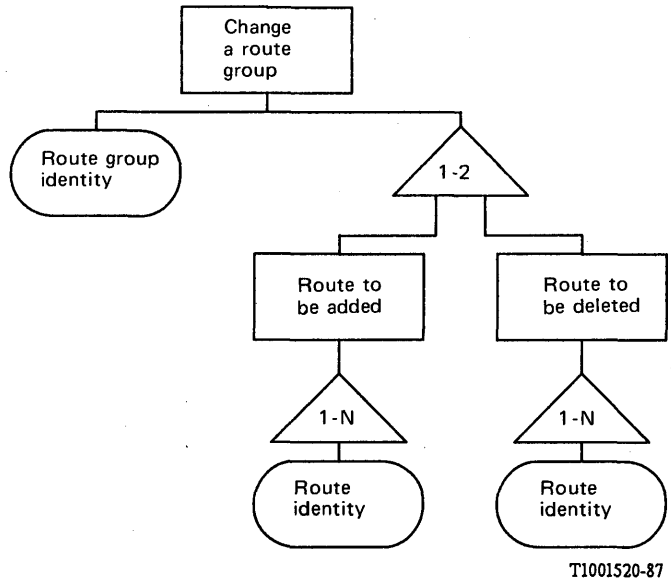
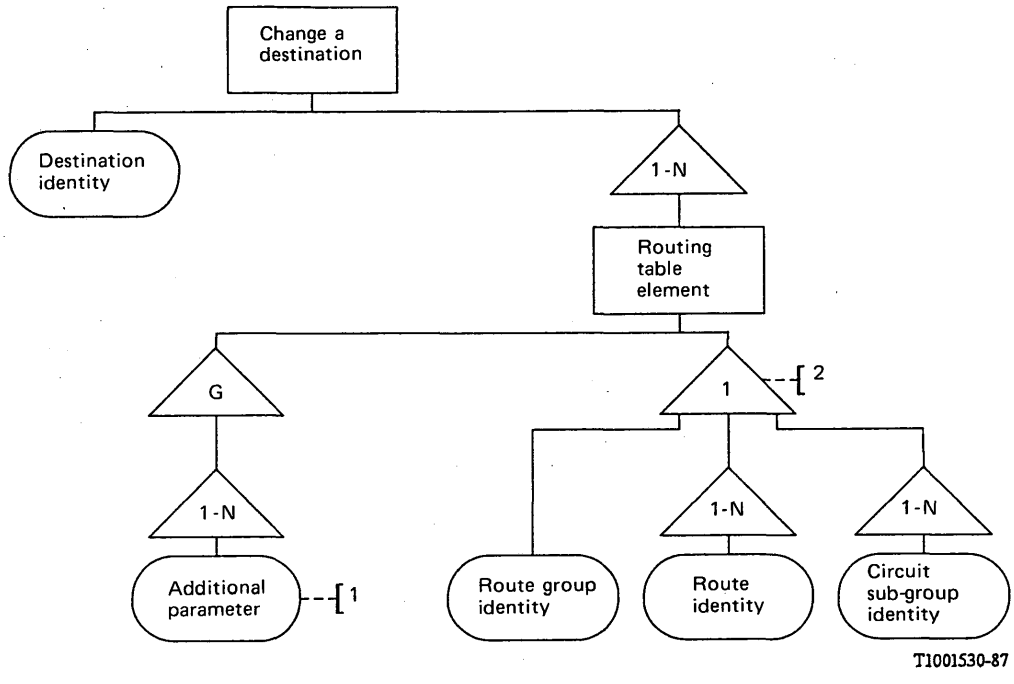


FIGURE B-19/Z.335

Change a route group



Note 1 – Possible parameters are:
- Calling party category.
- Class of incoming circuit sub-group.
- Parameters for network management.

Note 2 – The selection is made according to the option (A, B or C) adopted.

FIGURE B-20/Z.335

Change a destination

Recommendation Z.336

TRAFFIC MEASUREMENT ADMINISTRATION

1 General

This Recommendation has been developed in accordance to the methodology defined in Recommendations Z.332 and Z.333.

The main part of this Recommendation deals with the model of Traffic Measurement Administration and a glossary of the terms used is also included.

The list of operator jobs and the list of system functions to be controlled are contained in Annex A.

For each system function to be controlled by means of MML, one or more MML functions can be derived and each of them can be described using the metalanguage defined in Recommendation Z.333, in order to detail the relevant information structure.

Annex B contains a list of MML functions and information structure diagrams associated to each of them to be used as guidelines.

2 Introduction

Traffic measurement administration functions are related to data production, collection and output.

These data are achieved by means of periodic and non-periodic traffic measurements carried out in the telecommunications systems and are output by the systems in a suitable form.

The traffic measurement result outputs should contain the measurement results and general information about the measurement itself and about the system which performed the measurement, in order to ease the results analysis. Moreover, they should contain information summarizing the production of output blocks for checking purposes.

The traffic measurement model in § 4 is based upon a more general measurement model given in § 3.

3 General measurement model

A measurement is identified by three basic elements: time, entities, objects.

Time includes all the necessary information to define the start, the duration and periodicity of a certain measurement.

Entities describe the quantities for which data collection must be performed with a certain measurement, e.g. traffic flow, number of call attempts, congestion time.

Objects are intended as individual items within each object type on which the measurements are performed. Examples of object types are subscriber lines, circuits, circuit groups, elements of switching networks, geographical areas with their corresponding dialled code. The definition of measurements is based on an abstract model which contains the definition of a measurement matrix (see Figure 1/Z.336) in which each row represents one uniquely definable entity, e.g. number of call attempts, and each column represents a uniquely definable object type, e.g. incoming junction group (see Figure 2/Z.336).

A certain combination of entities and object types corresponds to certain entries in the measurement matrix and forms a measurement type. It is recognized that part of these measurement types may be standardized while the rest of them seem to be system and/or administration dependent. It should be noted that some of the entries in the measurement matrix could be impossible (e.g. call congestion on an incoming trunk) and some others could be more or less meaningless. A single object is defined by its type and/or its individual object identity. In some measurement types, the number of objects is fixed. In other types, one can choose for the actual measurement some or all of the allowed objects by means of MML commands. The chosen (selected) objects form an object list.

The structure of the division of object types and entities is open-ended, in such a way that any new object type or entity may be added.

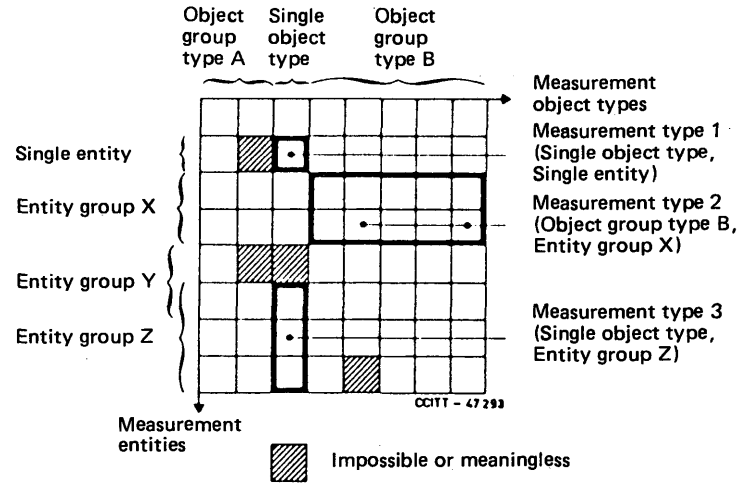


FIGURE 1/Z.336

Measurement matrix example

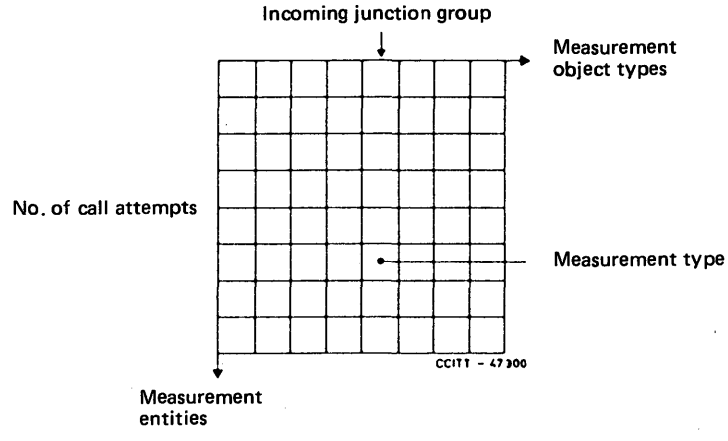


FIGURE 2/Z.336

Application of measurement matrix to a Traffic Measurement

4 Traffic measurement administration model

4.1 *Basic classes of measurements*

Two basic classes of measurements are envisaged (see Figure 3/Z.336). The first class (A) is measurement of undetermined duration while the second one (B) is intended to be performed only for a predetermined duration. The start of a measurement may be intended as instantaneous or delayed for a defined time duration, t_1, from the activation of the measurement. Since the stop time of a measurement of class A is not given when the measurement is activated or created, it has to be given during the measurement unless the measurement is intended to go on forever.

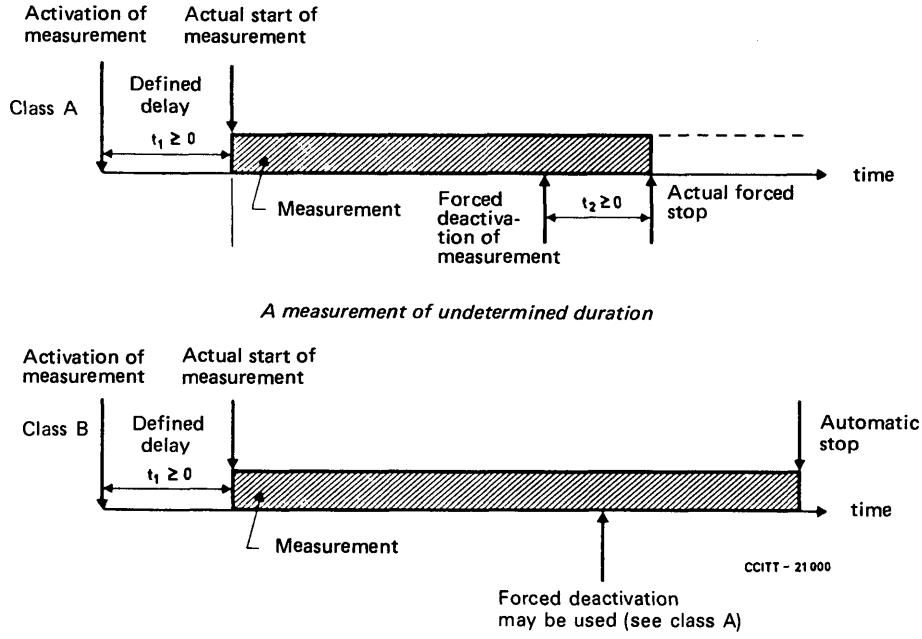


FIGURE 3/Z.336

Measurement duration classes

From the deactivation point of time there may be a defined delay of t_2 before the measurement is stopped. In the creation of a measurement a start time may optionally be provided, in which case for that particular measurement, the activation function is not necessary.

Time parameters needed to control a measurement can be divided into three groups:

1) measurement type dependent time parameters [interval parameters of a measurement type, e.g. sampling interval[1)]];
2) measurement dependent time parameters (e.g. time parameters which define the periodicity of measurement);
3) measurement independent time parameters (e.g. time parameters which are related to the actual start or stop of a certain measurement in activation and deactivation functions).

4.2 *Traffic measurement structure*

A traffic measurement (in the following called measurement) consists of:

- measurement set information,
- time information,
- output routing and scheduling information (output parameters).

Measurement set information, time information, output routing and scheduling information may be completely or partially pre-defined (initially provided by the supplier but changeable via MML inputs) or fixed (not changeable via MML inputs). The MML functions described for traffic measurements administration are intended to be supported to the extent that there is a need for user manipulation of the identified information items.

If some of this information is fixed in a system, the relevant MML functions may not be provided in that system.

[1)] Sampling interval, the time interval between two consecutive samplings.

4.2.1 *Measurement set information*

Measurement set information consists of one or several selected measurement types with defined objects (object lists) and measurement type dependent parameters (e.g. sampling interval, number of events of a certain category, destination codes, etc.).

Note that for traffic measurement administration purposes measurement types are fixed at a given moment in time and they cannot be created, deleted or changed by MML commands; these measurement types may be changed only later by supplier releases according to new requirements. It is recognized that administrations may require MML functions to administer measurement types, grouping predefined entities of object types. Such functions should be considered as system extension and upgrade functions and, therefore, they should belong to the system control functional area. However, due to the fact that system control functions will not be inserted in the present Recommendations, they are described hereafter.

4.2.2 *Time information*

Measurements of types A and B may involve continuous recording or recording on predetermined days (recording days).

For measurements performing continuous recording only the start data is needed.

For recording on predetermined days, these days are determined on a periodical basis (periodicity pattern) in case of measurements of undetermined duration. For measurements of predetermined duration, the recording days are determined on a periodical basis or on a non-periodical basis (dates of recording days). These possibilities are summarized in Figure 4/Z.336.

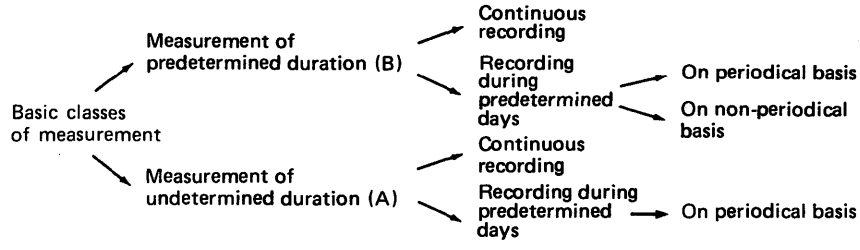


FIGURE 4/Z.336

Basic classes of measurement

Time data are defined at three main levels, as shown in Figure 5/Z.336.

Measurement level contains information about either:

- dates of recording days (in case of a non-periodical measurement). The start and stop date of the measurement are implicitly defined by the dates of the first and the last recording day. No activation function may be needed in this case;
- periodicity pattern (in case of periodical measurement) of recording and non-recording days.

Recording day level contains information about the start time and stop time for the recording periods within a recording day (e.g. from 09 to 12 and from 15 to 17). No overlap of recording periods is allowed for the same measurement.

Recording period level contains information about the periodicity of the data collection based on the result accumulation period. The result accumulation period is the time interval within a recording period during which the required measurement entities are processed and at the end of which results are stored for immediate or later output (e.g. 15 minutes). The result accumulation period can be shorter than the recording period; in that case more than one set of data is collected for each of the recording periods to be routed toward the output media according to the results output schedule.

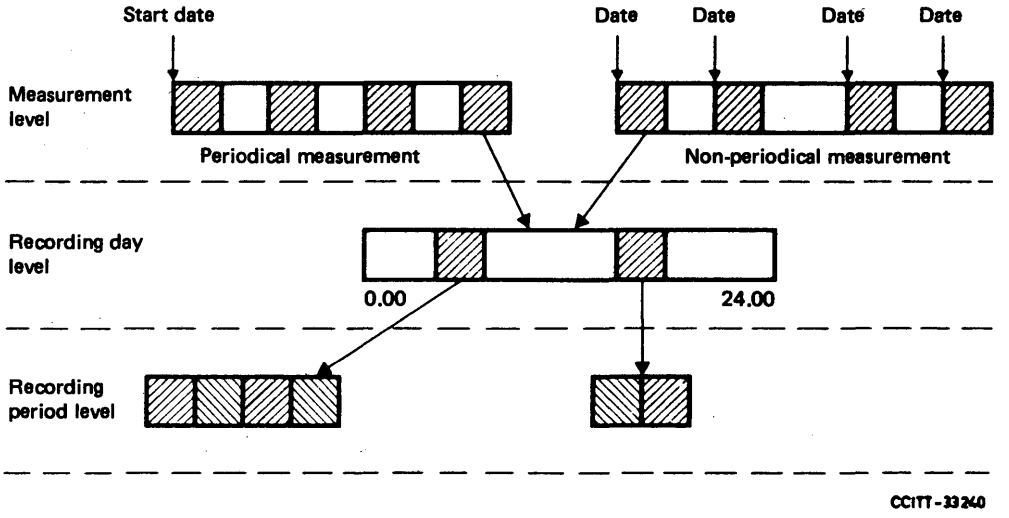


FIGURE 5/Z.336

Time information

5 Additional information

5.1 *Measurement output contents and procedures*

Activation of a traffic measurement causes the output of measurement results with the following procedures.

The produced output is routed toward the media specified in the output routing list associated to the measurement, e.g. printers, magnetic tapes, data links, system output files, etc. The output is made according to the output schedule.

The measurement results output is done according to time data related to the measurement. A measurement results output is made with the following logical blocks:

a) a "begin block" which contains measurement data, parameters i.e. measurement types data, time data, output data and data of interest related to the exchange configuration;

b) one or more "result blocks", one for each result output period, which contain the measurement results;

c) an "end block" which contains a general summary about the performance of the measurement, i.e. number of result blocks, number of interruptions of the measurement and the causes of the deactivation of the measurement (scheduled or forced).

If during the performance of the measurement, the measurement is suspended (e.g. due to a system crash) the measurement results output could be continued after the system restart with a new output of the begin block. This continuation may be accomplished automatically by the system or by user action. The system should notify the user via an output if the latter case applies.

The relationship between time data for the result accumulation period and time data defining the results output schedule is system or even measurement dependent and it is not considered herein.

5.2 *Simplification of traffic measurement administration*

It is recognized that, for particular applications, there may not be an interest in administering the data base of traffic measurements. Consequently, the only MML functions needed are activation and deactivation functions.

In such cases, in order to ease the operator's work, the association between the measurement and the objects may be made when activating the measurement itself, provided that the association is unambiguous.

6 Glossary of used terms

Recording

Performance of the operations implied by the measurement entities in order to collect the required data.

recording day

Day when a recording is performed. Several recording periods are allowed within a recording day. No overlap of recording periods is allowed for the same measurement. Each recording period can have a different length.

start date

Start day for the measurement execution.

stop date

Stop day for the measurement execution.

periodicity pattern

A pattern which indicates which days are recording (or results output) days and which are not. The start day positions this time span. Once activated, the execution of the measurements (or of the results output) is performed according to this pattern, until disabled by a deactivation command.

start time

Time for beginning the recording period in a recording day.

stop time

Time for terminating a recording period in a recording day.

recording period

A period of recording during a recording day.

results accumulation period

Time interval within a recording period during which the required measurement entities are processed and at the end of which results are stored for immediate or later output.

output parameters

Data determining output routing and scheduling.

results output routing

Data defining the media to which results output is to be directed.

results output schedule

Data specifying a set of days (or a periodicity pattern) and of times during these days when the output of the results is to be made.

ANNEX A

(to Recommendation Z.336)

List of system functions to be controlled by MML and list of jobs

A.1 *List of system functions to be controlled by MML*

1) Performing traffic measurements.
2) Scheduling traffic measurements execution and results output.
3) Managing measurements' data.
4) Retrieving measurements' data.

A.2 *List of jobs*

1) To create new measurements or measurement components and to modify old ones, by defining the entities to be measured and the objects and parameters of the measurements themselves (what and how to measure):
   - the purpose of this job is to create and/or modify a set of data which is used by the system to perform a measurement in a given way;
   - the system is supposed to record the set of data of the measurement, and to check their static correctness;
   - the user is supposed to input/change all relevant data. The modification of data may be performed by means of different procedures, depending whether or not those data are related to activated measurements;
   - the complexity of the job could be high depending on the amount of data to be input;
   - the frequency of the job is low;
   - the job is supposed to be performed at exchange and/or OMC level.
2) To delete obsolete measurements or measurement components:
   - the purpose of the job is to delete measurement of no further use or measurement components to release the employed resources;
   - the system is supposed to delete the data related to a specified measurement if the measurement is not active. The system is supposed to delete a measurement component only if it is not an active measurement component;
   - the user is supposed to input the identities of measurements or measurement components to be deleted;
   - the complexity of the job is low;
   - the frequency of the job is low;
   - the job is supposed to be performed at exchange and/or OMC level;
3) To define the measurement results output routing and scheduling (where and when the results will be output):
   - the purpose of the job is to define where the measurement outputs have to be routed to and when they should be output;
   - the system has to route the measurement outputs toward the recording media or toward other systems specified, according to the results output schedule;
   - the user has to input the identity of the destination of the output and the results output schedule to be followed by the system;
   - the complexity of the job is low;
   - the frequency of the job is medium;
   - the job may be performed at exchange and/or OMC level.
4) To activate and to deactivate measurements (when to measure):
   - the purpose of the job is to activate and/or deactivate the performance of the measurements that have been previously defined;
   - the system is supposed to activate and/or deactivate a measurement and to start the production of the results;
   - the user is supposed to input the date and time of activation and/or deactivation;
   - the complexity of the job is low;
   - the frequency of the job is medium;
   - the job may be performed at exchange and/or OMC level.

5) To retrieve different kinds of information related to traffic measurements:
   - the purpose of the job is to get information on measurements previously input in the system(s) in order to be aware of the current situation;
   - the system is supposed to output in suitable formats and on the selected device(s) the information requested;
   - the user is supposed to input the identity of the items to be interrogated and to select retrieving criteria;
   - the complexity of the job is low;
   - the frequency of the job is medium;
   - the job may be performed at exchange and/or OMC level.

ANNEX B

(to Recommendation Z.336)

Guidelines for the list of MML functions and associated information structure diagrams

B.1 *Introduction*

This Annex contains guidelines for the list of MML functions and associated information structure diagrams related to the routing administration model defined in Recommendation Z.336, § 4.

B.2 *List of MML functions*

This list contains possible MML functions for traffic measurement administration. Those functions dealing with information (e.g. measurement set, time data list, etc.) that is fixed in a system are not relevant for that system.

This list is not mandatory nor complete, it may vary according to Administration needs, telecommunication network levels, regulatory needs, etc.

These MML functions do not represent the actual command structure of any real implementation of the man-machine interface. Each of the MML functions identified can be implemented by providing one or more separate distinctive commands or several MML functions could be implemented by using a single command.

1) *Creation*
   - create a measurement;
   - create a measurement set;
   - create an object list;
   - create a time data list;
   - create an output routing list;
   - create a result output schedule.
2) *Deletion*
   - delete a measurement;
   - delete a measurement set;
   - delete an object list;
   - delete a time data list;
   - delete an output routing list;
   - delete a result output schedule.
3) *Activation*
   - activate a measurement.
4) *Deactivation*
   - deactivate a measurement.

5) *Interrogation*
   - interrogate a measurement;
   - interrogate a measurement set;
   - interrogate a mesurement type;
   - interrogate an object list;
   - interrogate a time data list;
   - interrogate an output routing list;
   - interrogate a result output schedule.

6) *Changing*
   - change a measurement;
   - change a measurement set;
   - change an object list;
   - change a time data list;
   - change an output routing list;
   - change a result output schedule.

7) *Administration of measurement types*
   - create a measurement type;
   - delete a measurement type;
   - change a measurement type.

B.3 *Information structure diagrams*

The information entities needed for the MML functions previously defined have been identified and are reported in this section by means of diagrams representing each MML function information structure (Figures from B-2/Z.336 to B-41/Z.336). In particular, the information structure diagrams of the measurement outputs are given in Figures from B-42/Z.336 to B-45/Z.336.

An overview of the measurement data structure is also given in Figure B-1/Z.336.

The metalanguage used is defined in Recommendation Z.333.

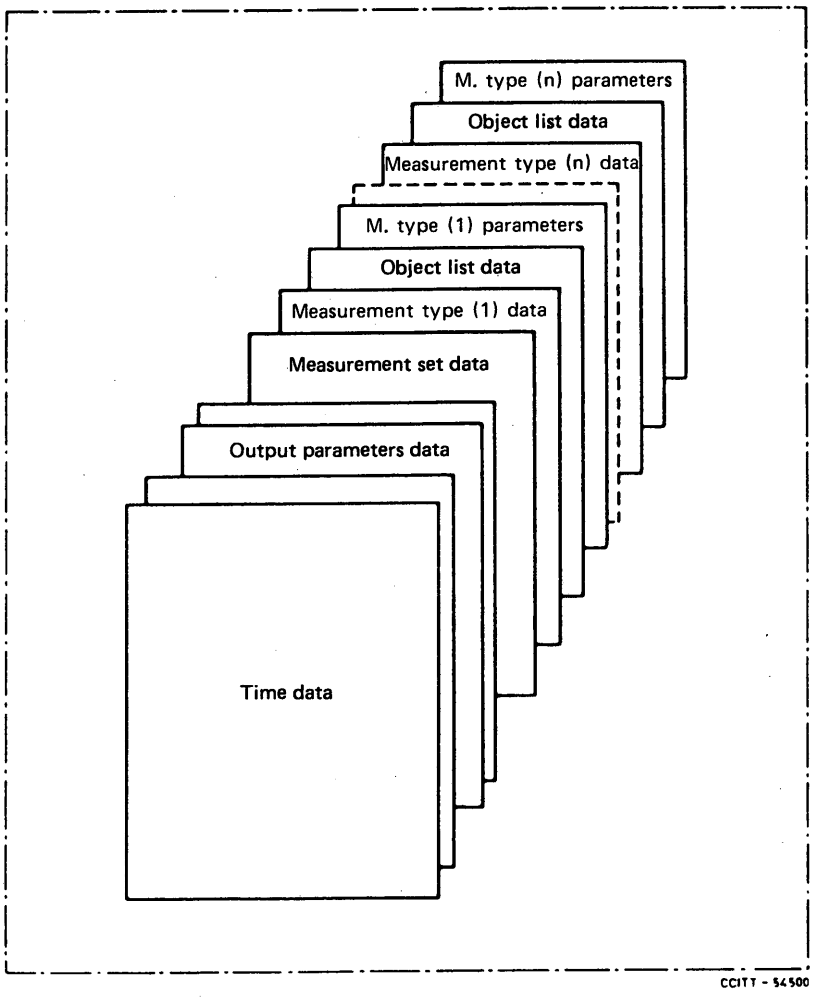


FIGURE B-1/Z.336

Traffic measurement data overview

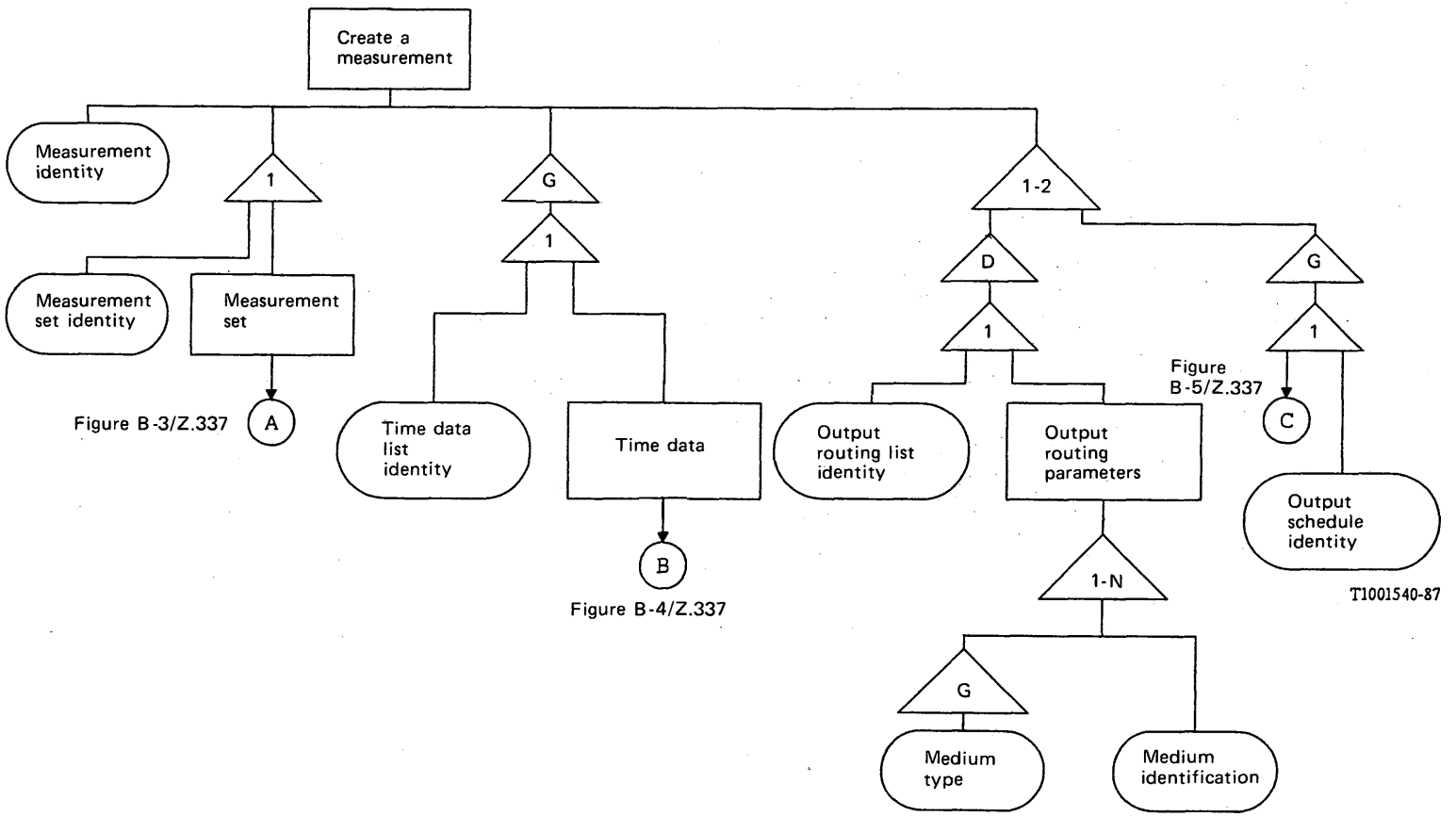
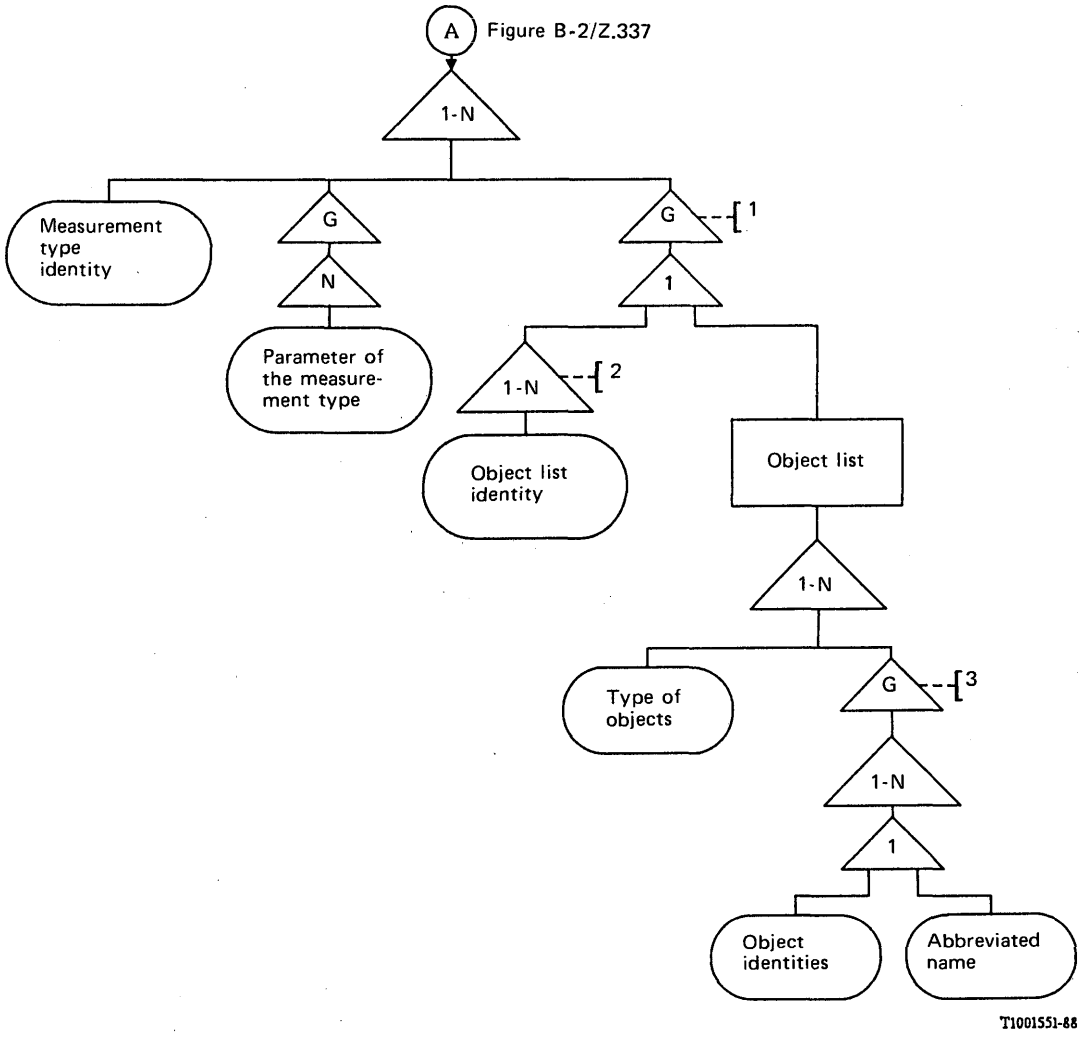


FIGURE B-2/Z.336

Create a measurement



Note 1 – No object list needed if the measurement type implies global measurements on a certain object type.

Note 2 – Multiple object lists imply a resulting merged list.

Note 3 – Zero is meaningful only for the measurement types implying global measurements on selectable object types.

FIGURE B-3/Z.336

Create a measurement (continued)

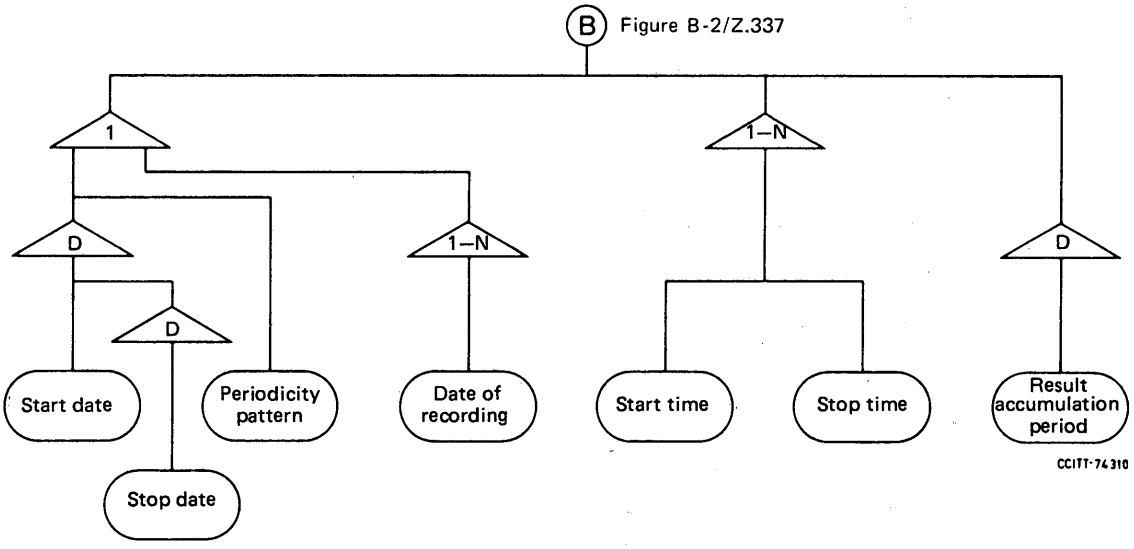


FIGURE B-4/Z.336

Create a measurement (Continued)

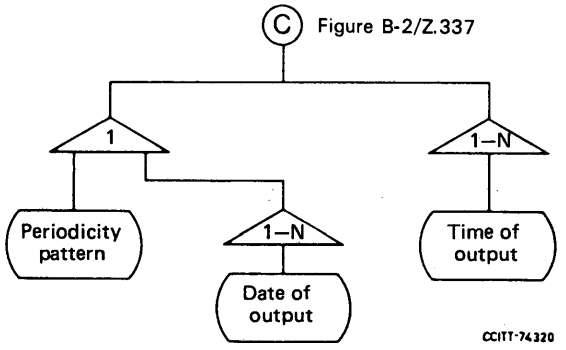


FIGURE B-5/Z.336

Create a measurement (Continued)

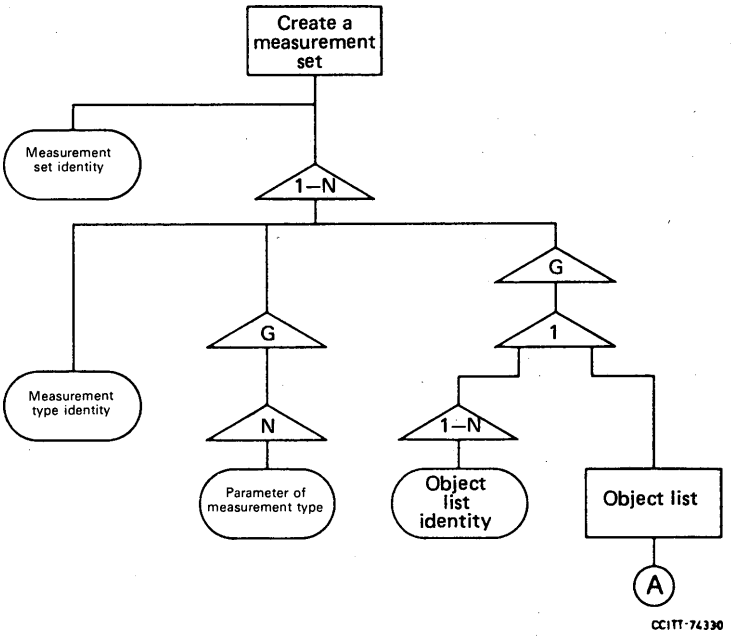


FIGURE B-6/Z.336

Create a measurement set

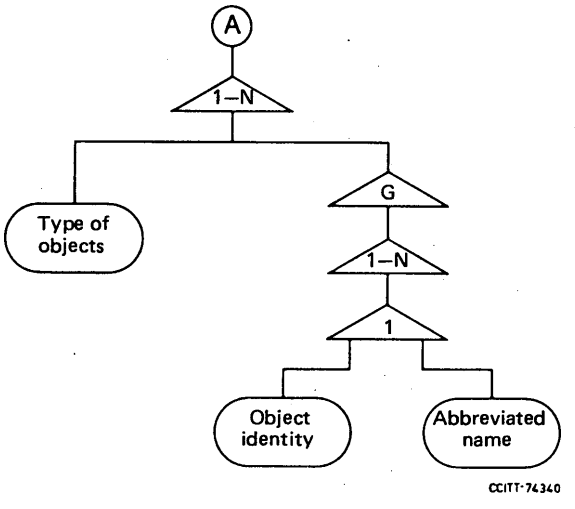


FIGURE B-7/Z.336

Create a measurement set (Continued)

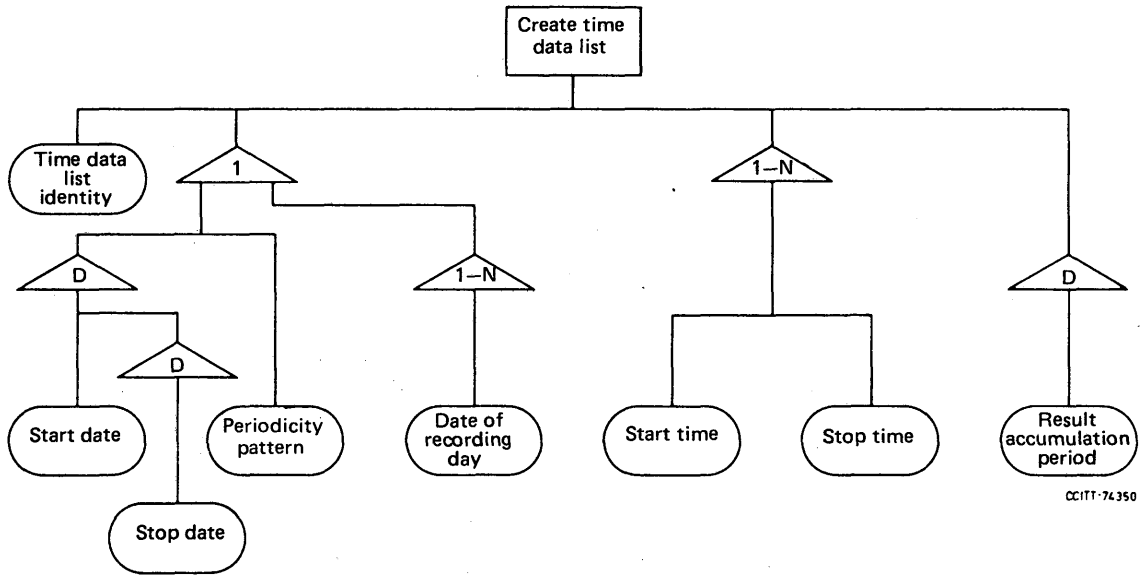


FIGURE B-8/Z.336

Create a time data list

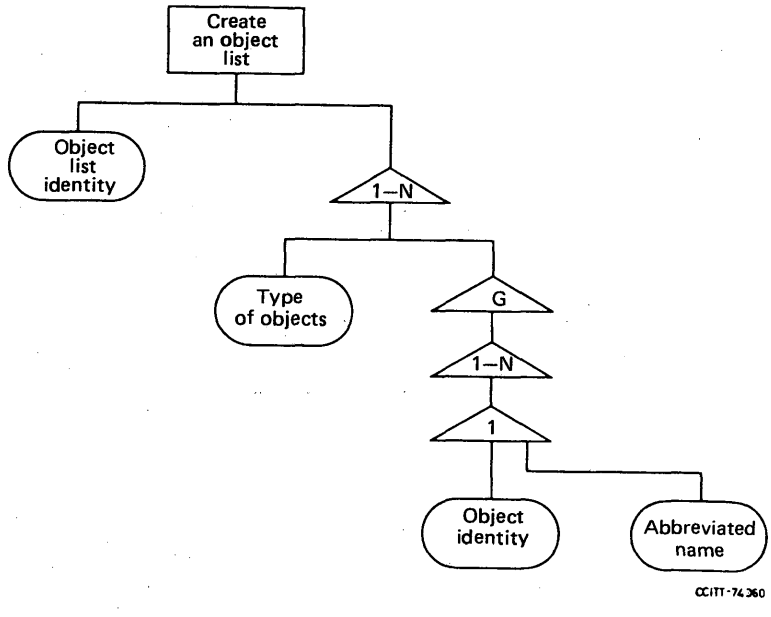


FIGURE B-9/Z.336

Create an object list

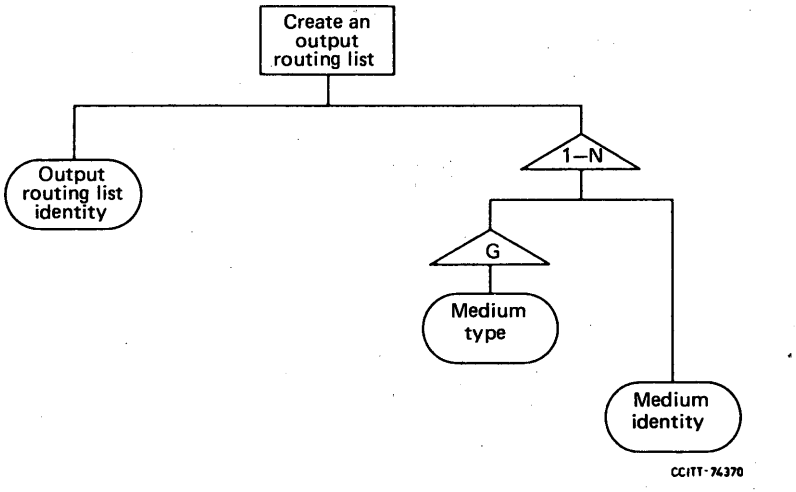
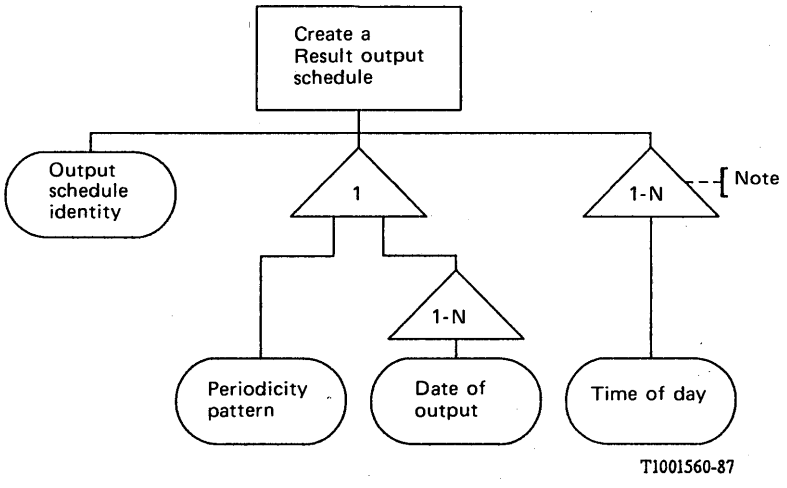


FIGURE B-10/Z.336

Create an output routing list



Note – Set of times may depend on output day.

FIGURE B-11/Z.336

Create a result output schedule

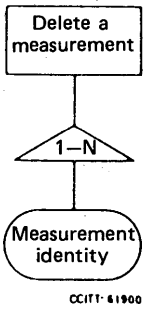


FIGURE B-12/Z.336

Delete a measurement

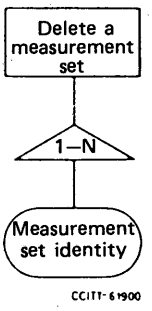


FIGURE B-13/Z.336

Delete a measurement set

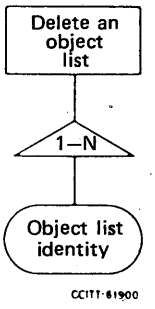


FIGURE B-14/Z.336

Delete an object list

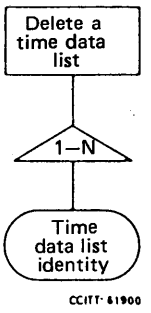


FIGURE B-15/Z.336

Delete a time data list

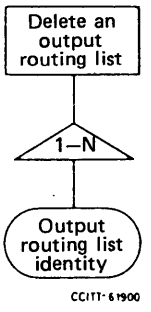


FIGURE B-16/Z.336

Delete an output routing list

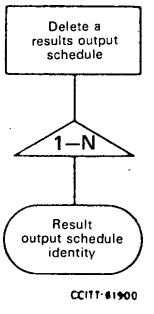
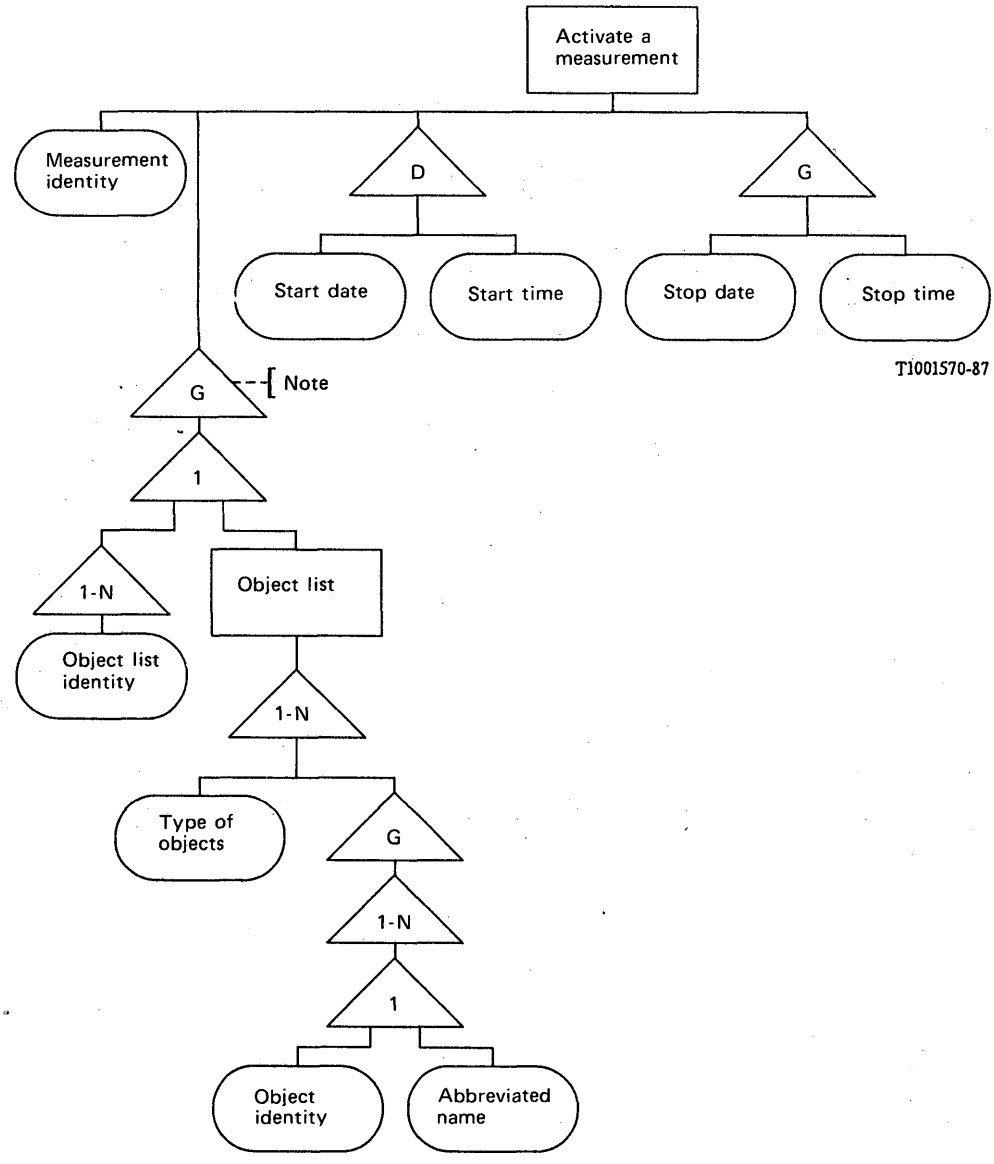


FIGURE B-17/Z.336

Delete a results output schedule



Note – The optionality does not apply when a simplified set of MML functions is used as described in Section 5.2.

FIGURE B-18/Z.336

Activate a measurement

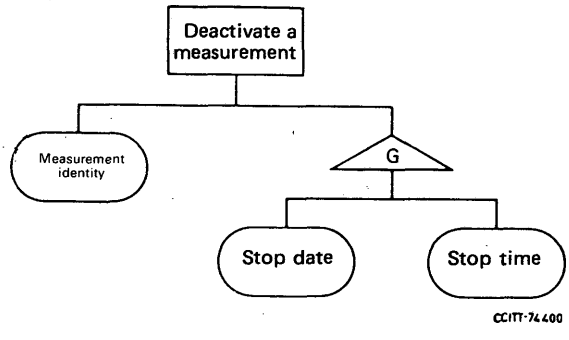
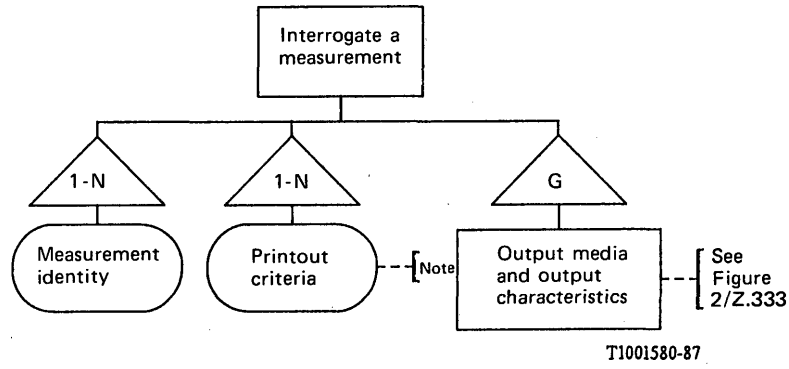


FIGURE B-19/Z.336

Deactivate a measurement

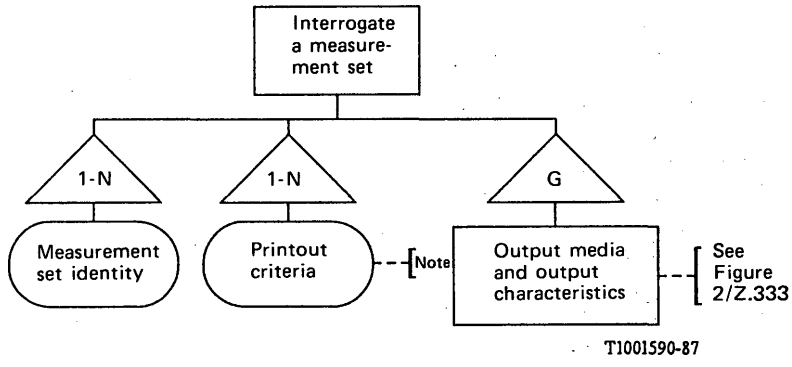


Note – Possible parameter values:

- object list,
- object list identity,
- measurement types,
- parameters of measurement types,
- measurement set,
- measurement set identity,
- time data,
- time data list identity,
- output routing list,
- output routing list identity,
- output schedule,
- output schedule identity,
- status (activated or not activated).

FIGURE B-20/Z.336

Interrogate a measurement

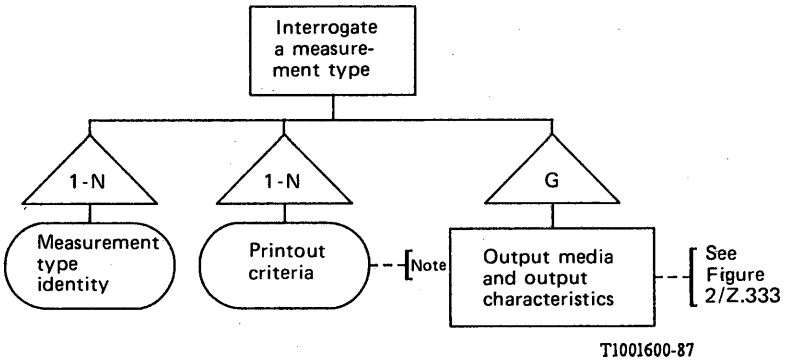


Note – Possible parameter values:

- measurement type identities,
- parameters and associated values,
- object list,
- measurements utilizing the identified set.

FIGURE B-21/Z.336

Interrogate a measurement set

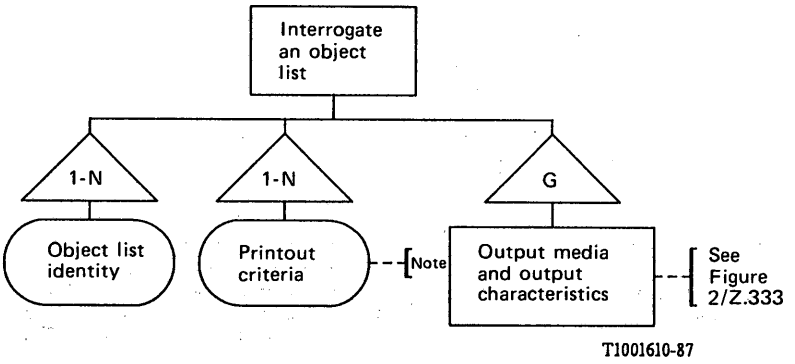


Note – Possible parameter values:

- list of parameters of the measurement type,
- object lists associated with the measurement type,
- sets utilizing the measurement type,
- measurements utilizing the measurement type.

FIGURE B-22/Z.336

Interrogate a measurement type

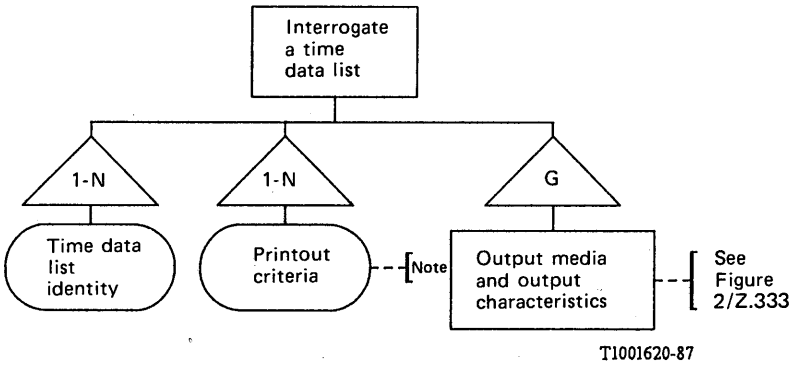


Note – Possible parameter values:

- object type,
- object type and individual object identities,
- measurements utilizing the object list.

FIGURE B-23/Z.336

Interrogate an object list

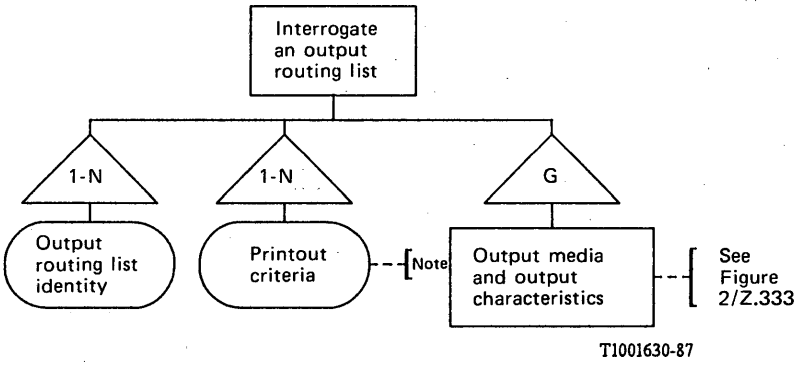


Note – Possible parameter values:

– time data,
– measurements utilizing the time data list.

FIGURE B-24/Z.336

Interrogate a time data list

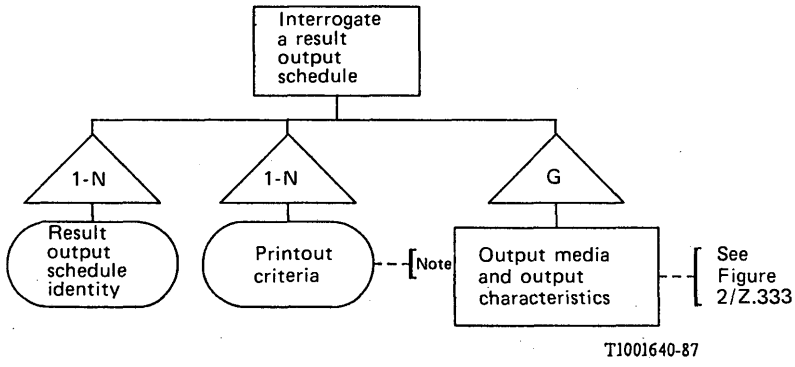


Note – Possible parameter values:

– output routing data,
– measurements utilizing the output routing data list.

FIGURE B-25/Z.336

Interrogate an output routing list



Note – Possible parameter values:

– results output schedule data,
– measurements utilizing the results output schedule.

FIGURE B-26/Z.336

Interrogate a results output schedule

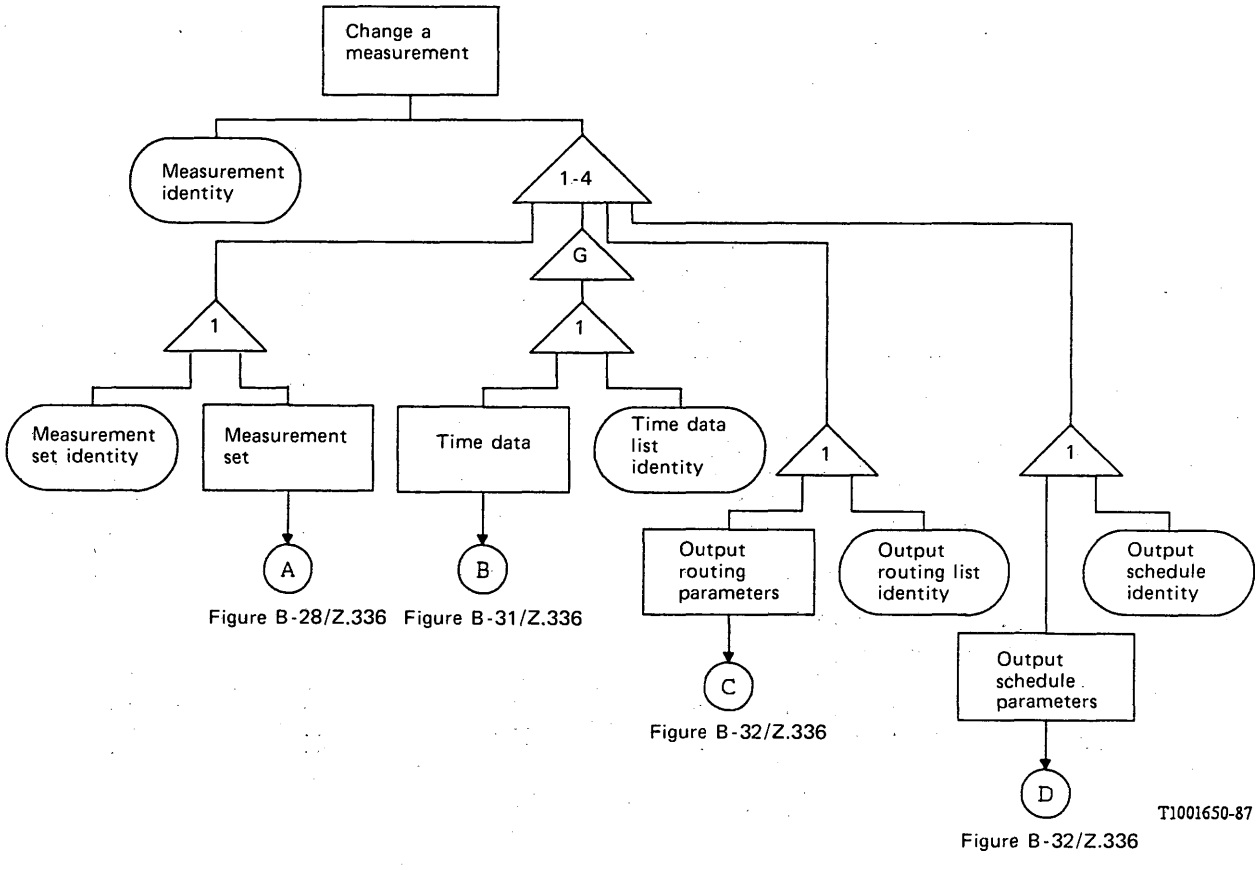


FIGURE B-27/Z.336

Change a measurement

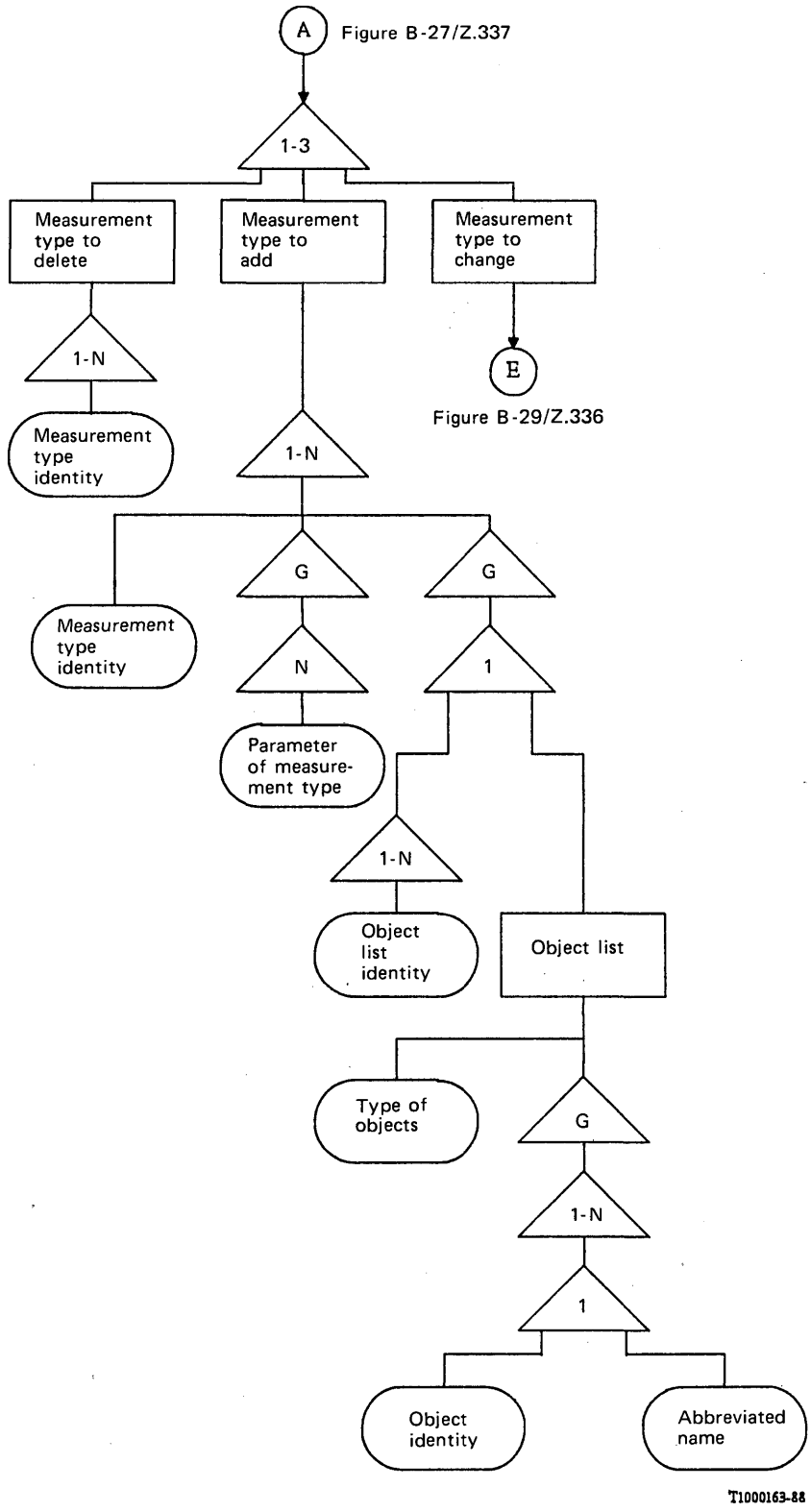


FIGURE B-28/Z.336

Change a measurement (continued)

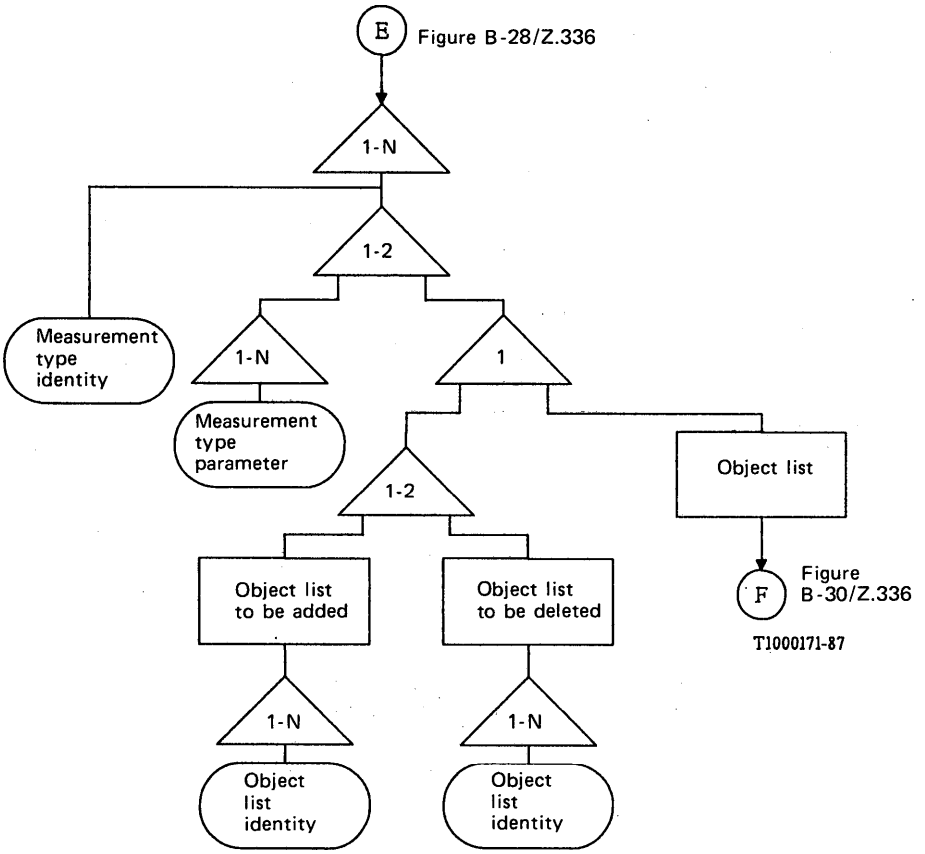


FIGURE B-29/Z.336

Change a measurement (continued)

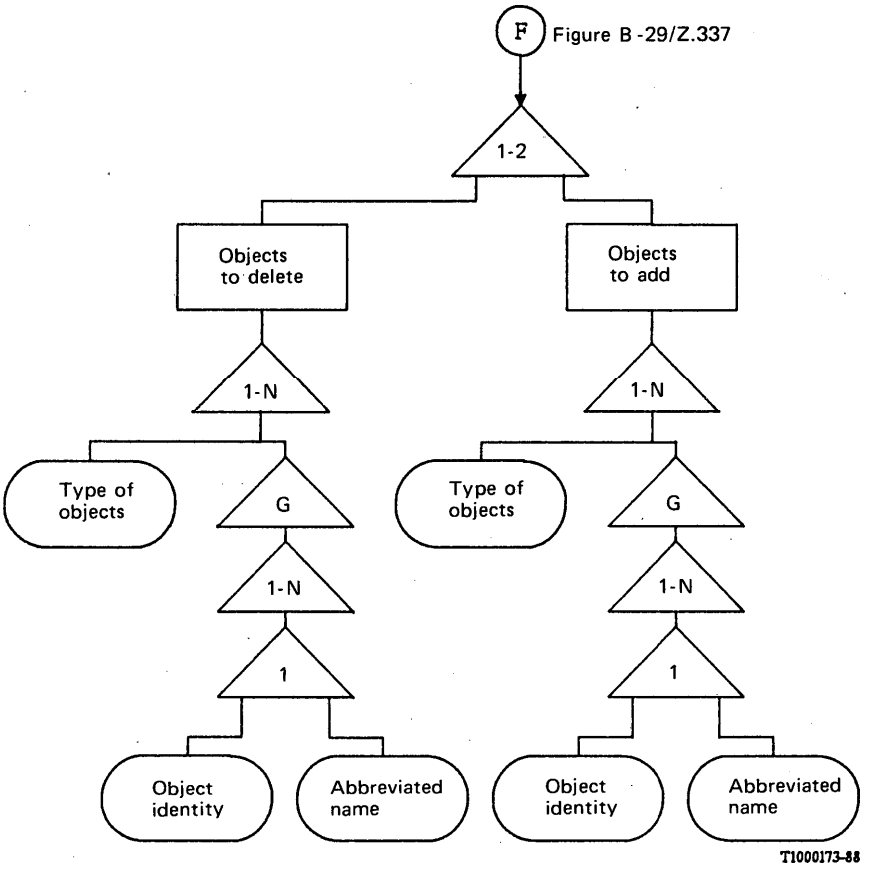


FIGURE B-30/Z.336

Change a measurement (continued)

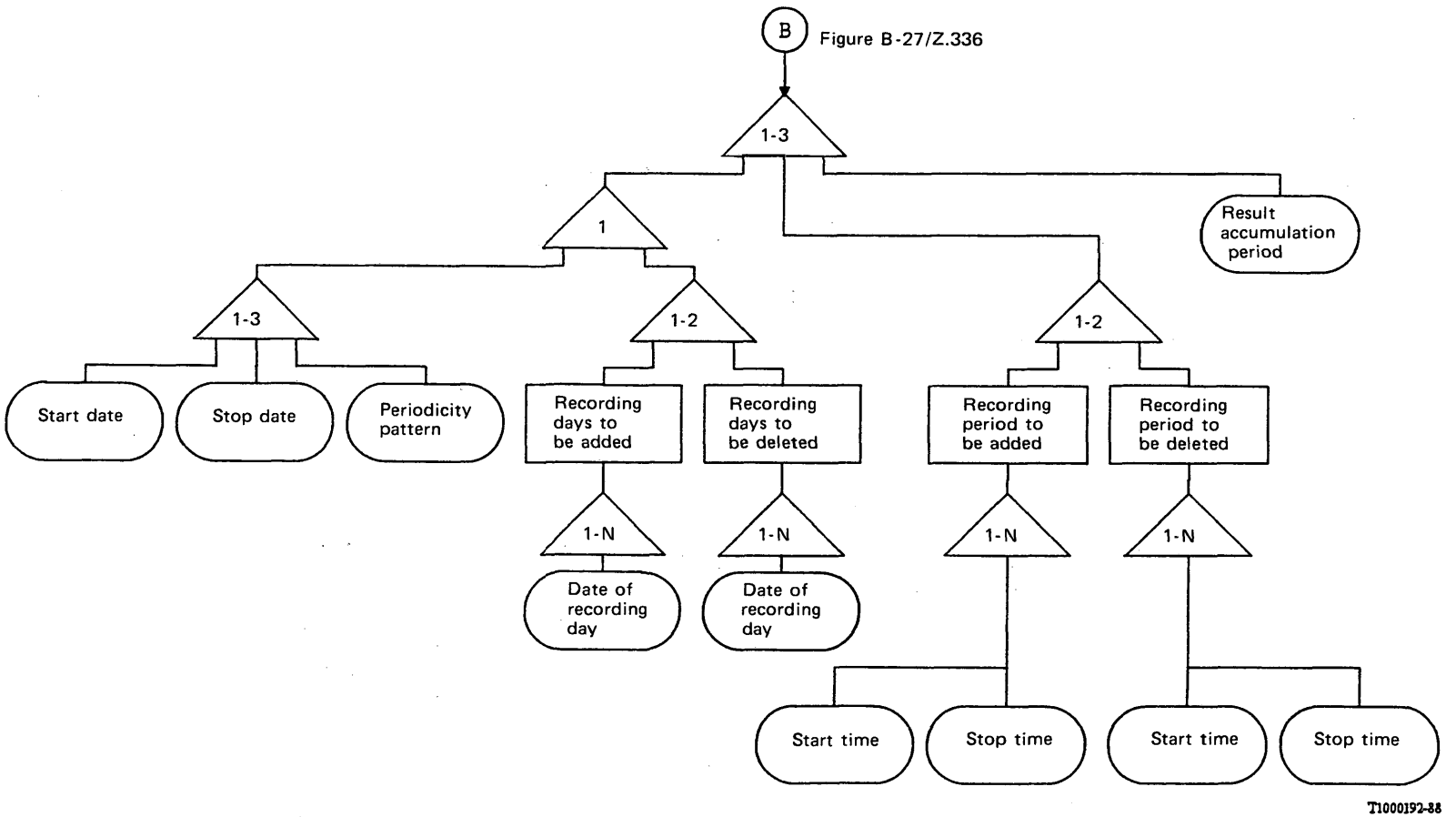


FIGURE B-31/Z.336

Change a measurement (continued)

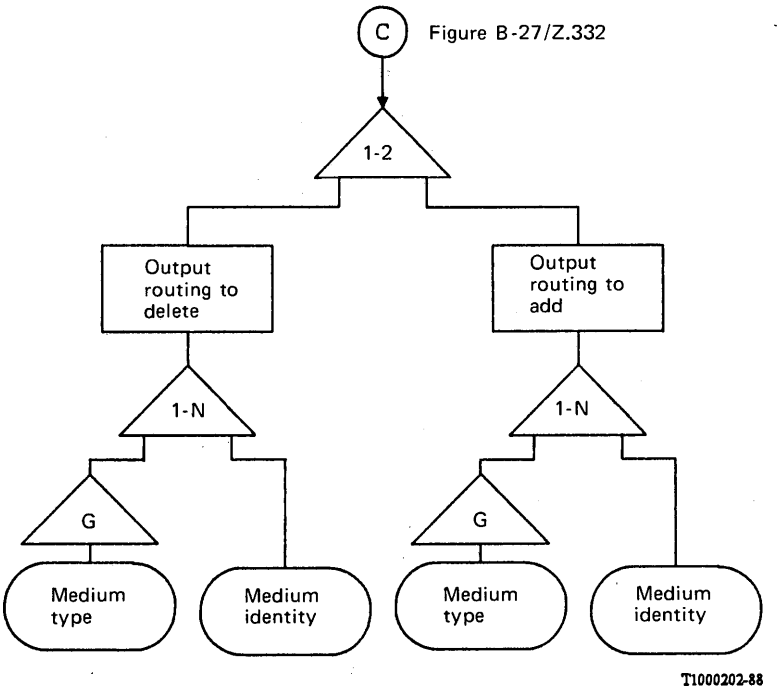
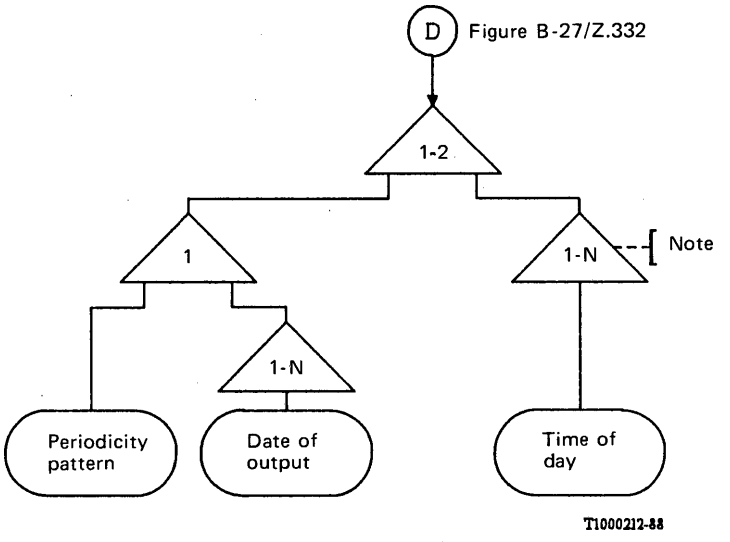


FIGURE B-32/Z.336 (Sheet 1 of 2)

Change a measurement (continued)



Note – Set of times may depend on output day.

FIGURE B-32/Z.336 (Sheet 2 of 2)

Change a measurement (continued)

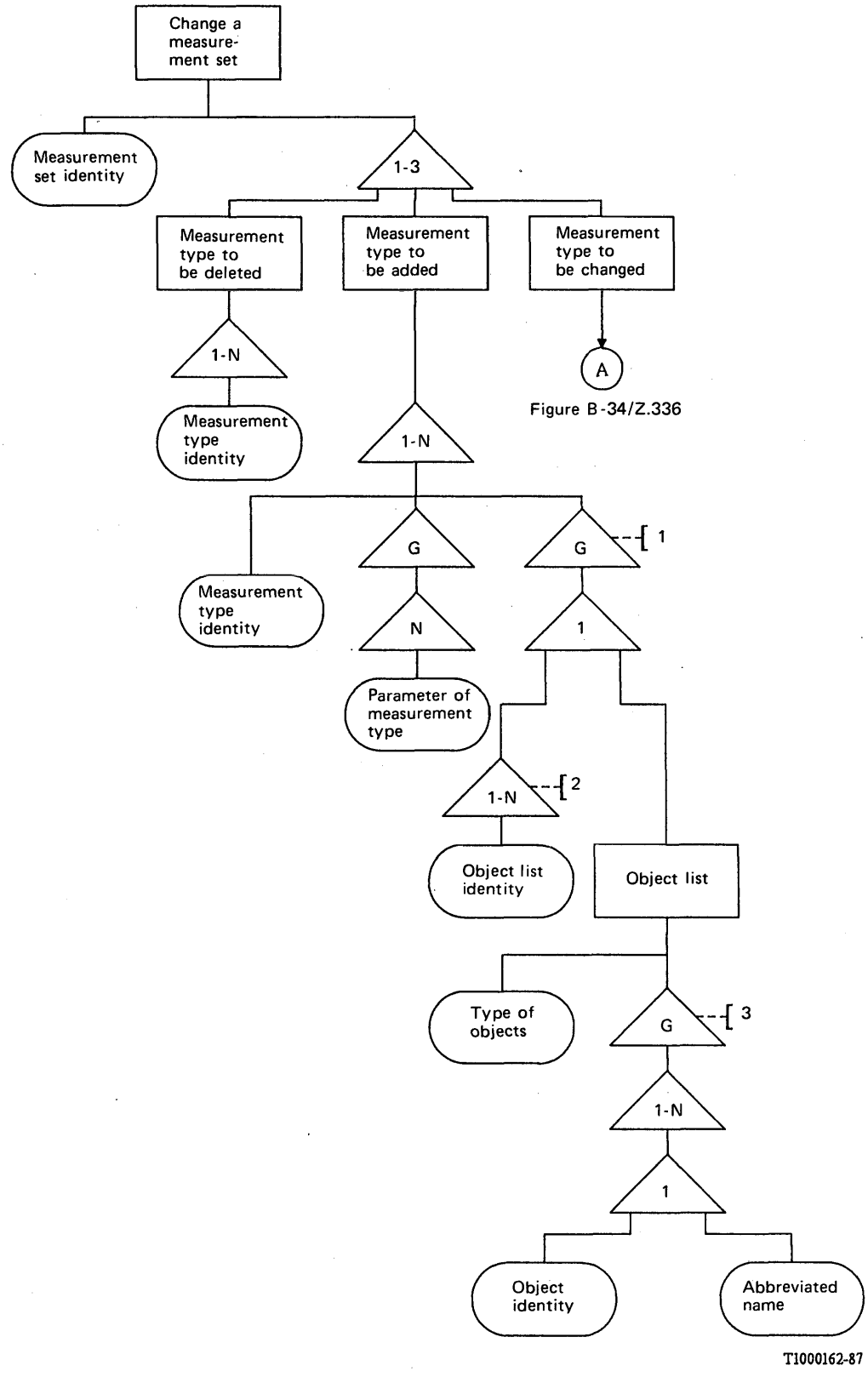


Figure B-34/Z.336

Note 1 – No object list is needed if the measurement type implies global measurements on a certain object type.

Note 2 – Multiple object lists imply a resulting merged list.

Note 3 – Zero is meaningful only for the measurement types implying global measurements on selectable object types.

FIGURE B-33/Z.336

Change a measurement set

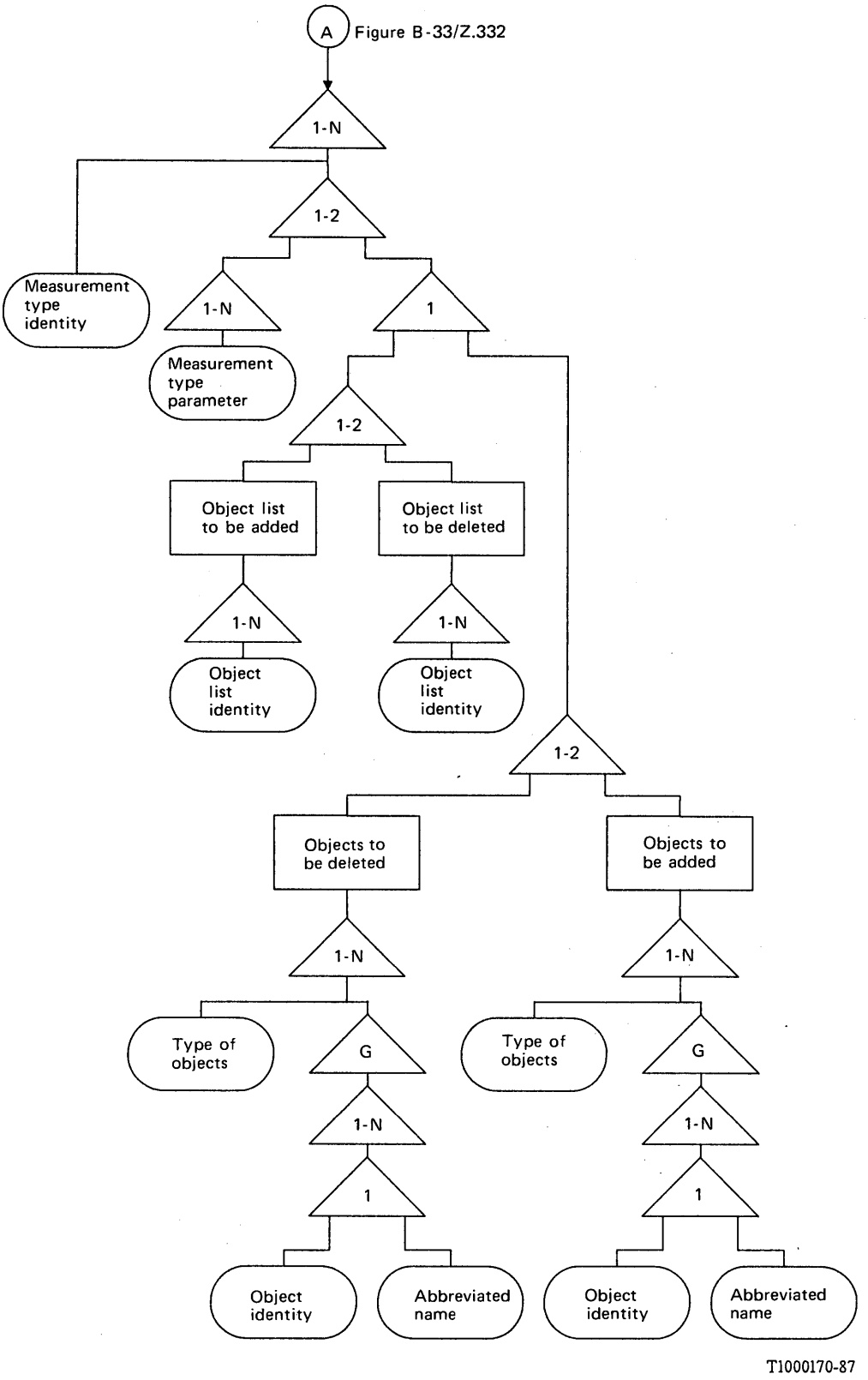


FIGURE B-34/Z.336

Change a measurement set (continued)

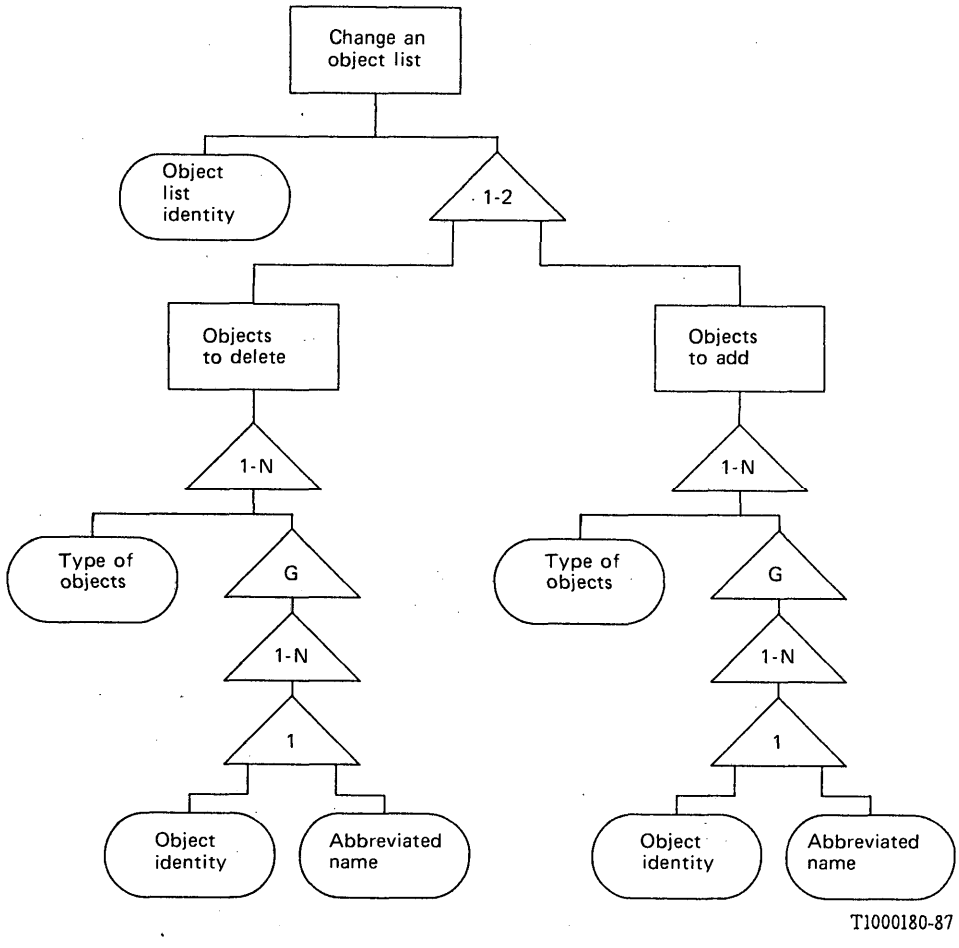


FIGURE B-35/Z.336

Change an object list

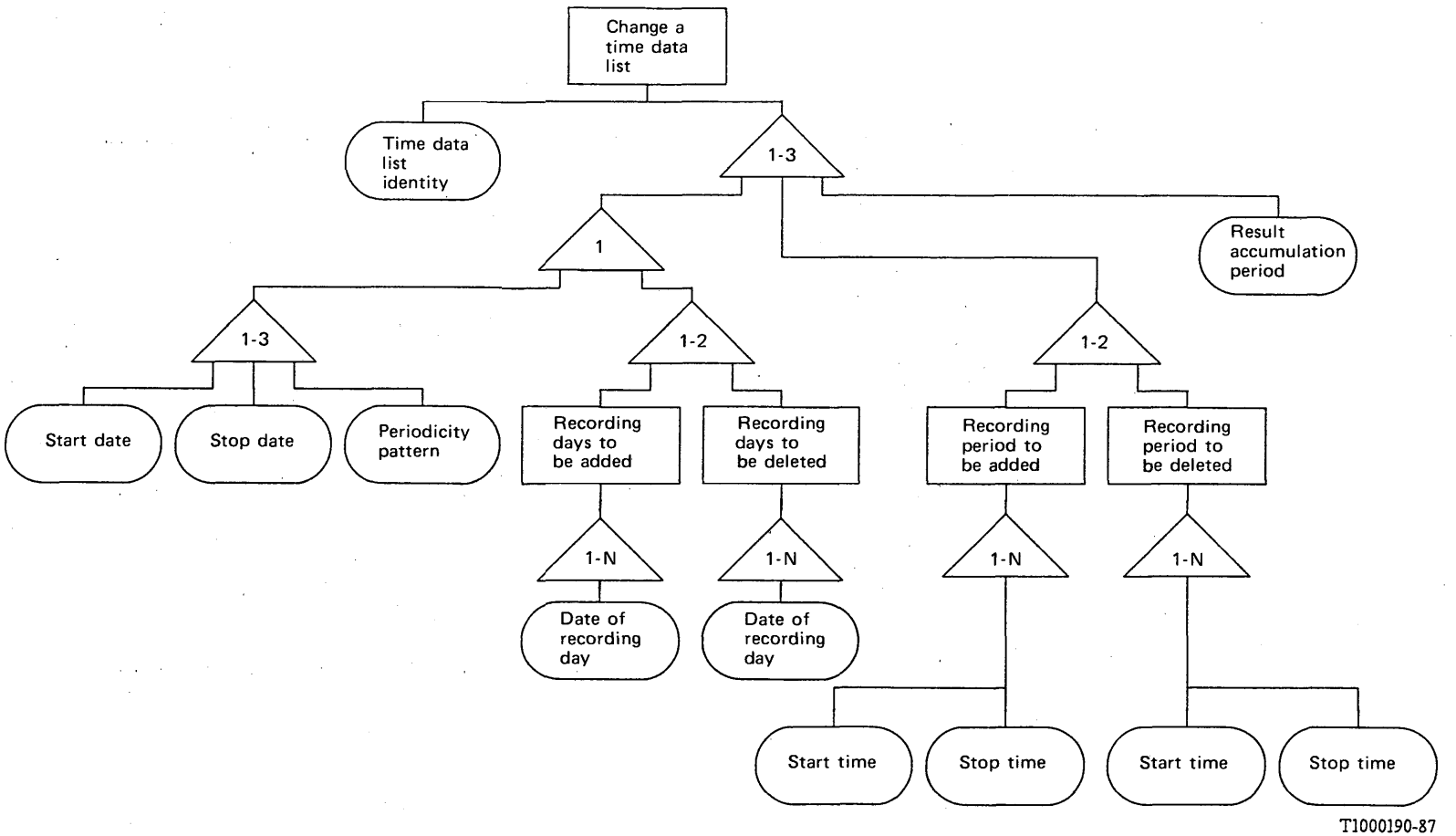


FIGURE B-36/Z.336

Change a time data list

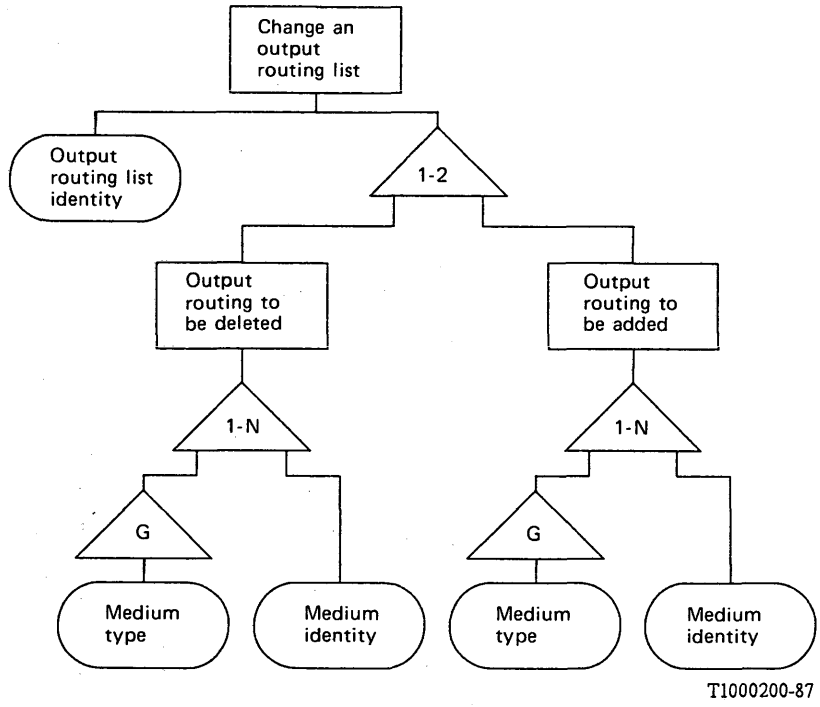
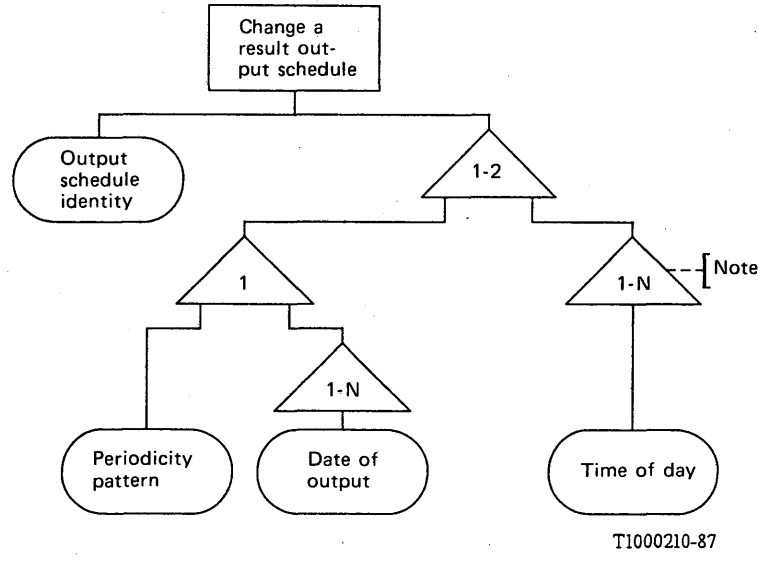


FIGURE B-37/Z.336

Change an output routing list



Note – Set of times may depend on output day.

FIGURE B-38/Z.336

Change a result output schedule

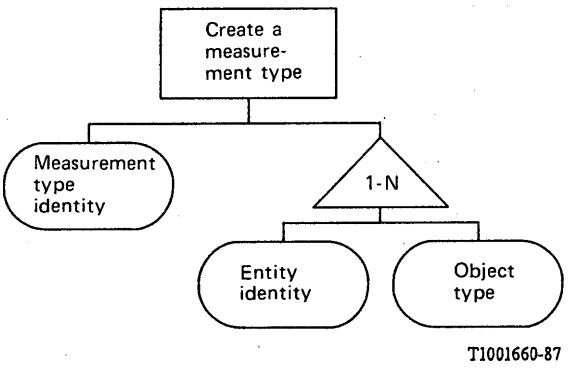


FIGURE B-39/Z.336

Create a measurement type

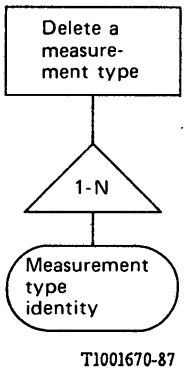


FIGURE B-40/Z.336

Delete a measurement type

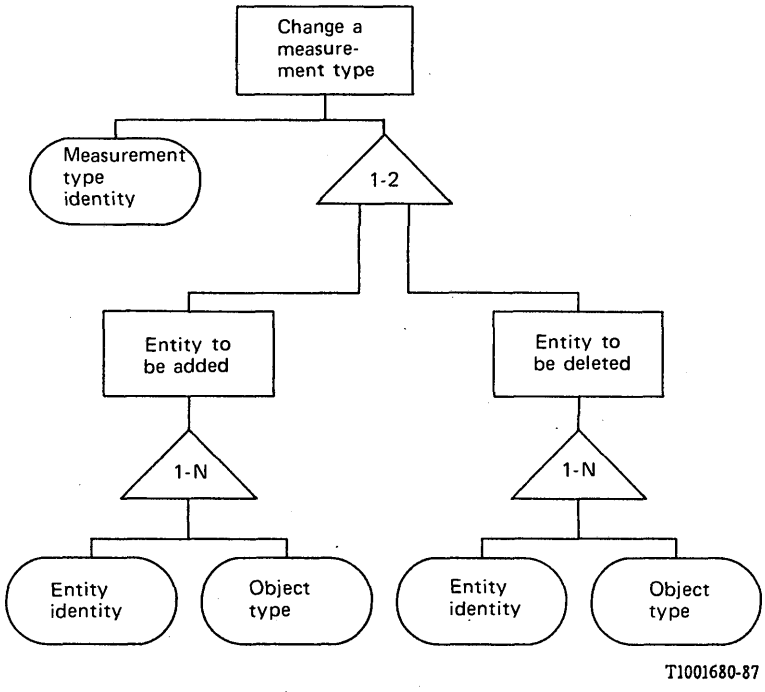


FIGURE B-41/Z.336

Change a measurement type

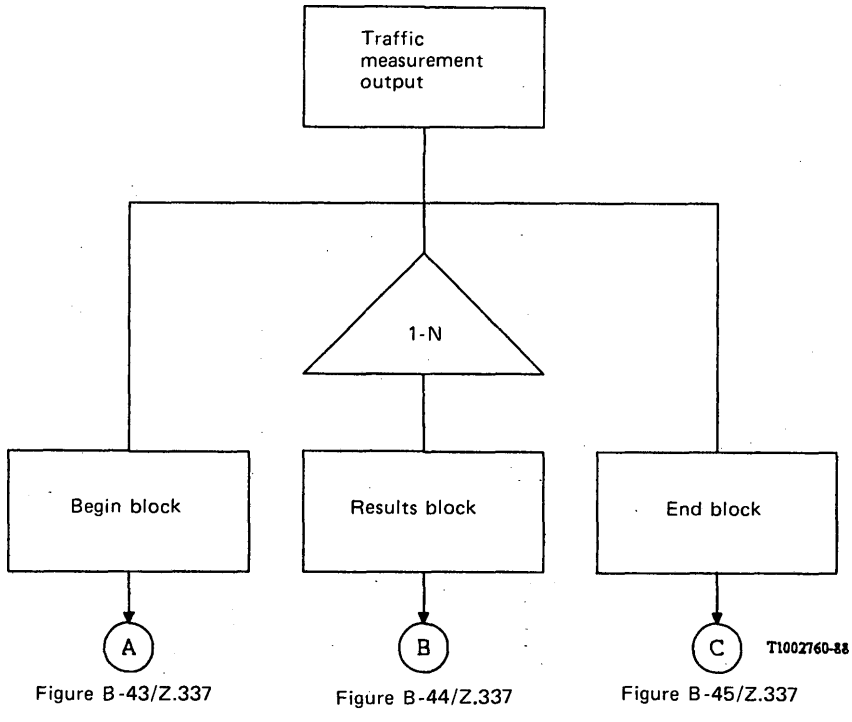
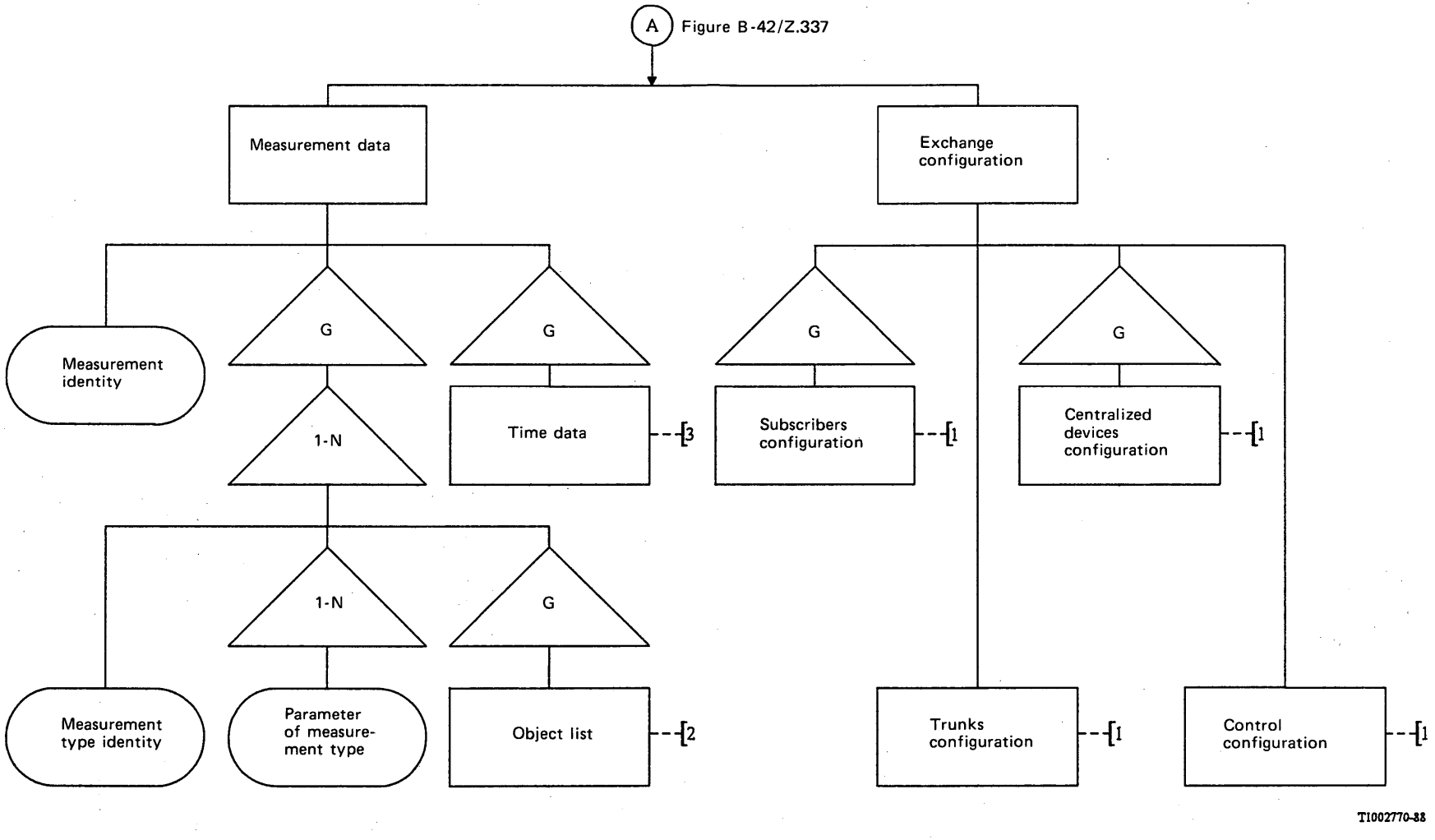


FIGURE B-42/Z.336

Traffic measurement output



Note 1 – Not expanded further.

Note 2 – See Figure B-9/Z.336.

Note 3 – See Figure B-8/Z.336.

FIGURE B-43/Z.336

Traffic measurement output (continued)

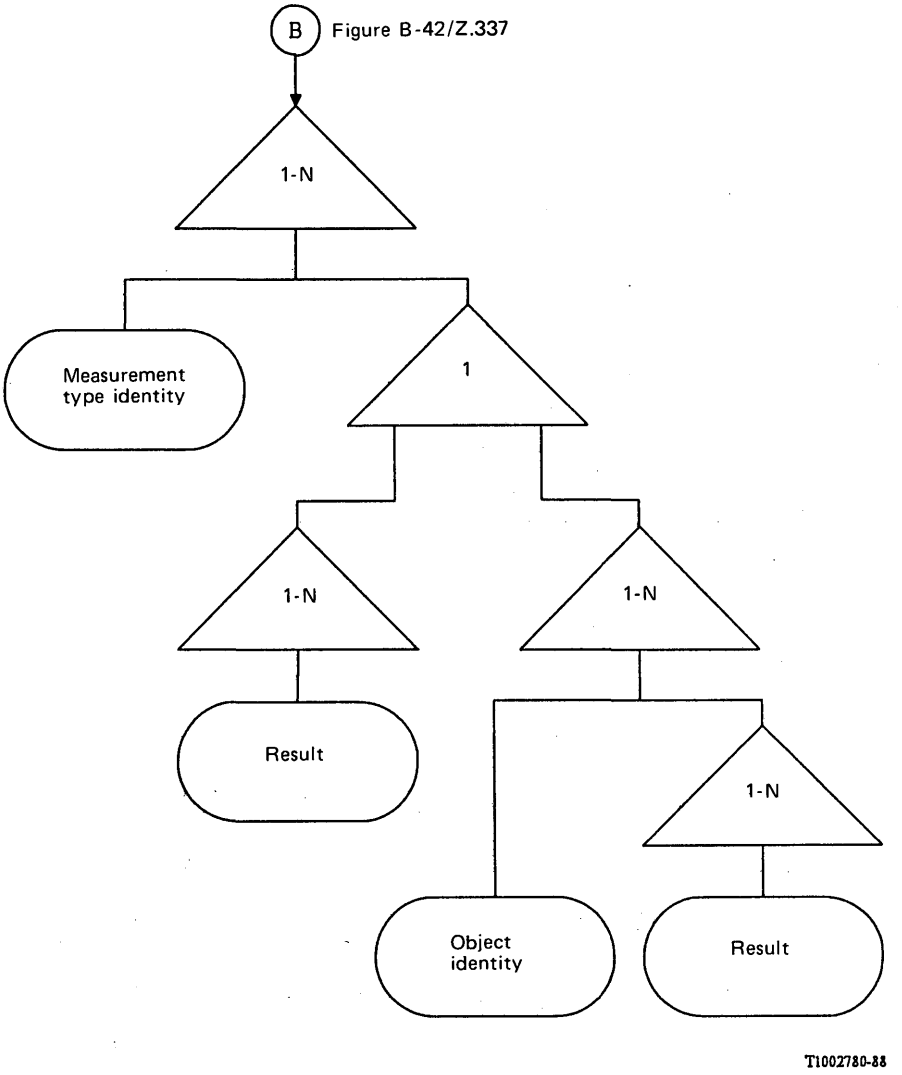


FIGURE B-44/Z.336

Traffic measurement output (continued)

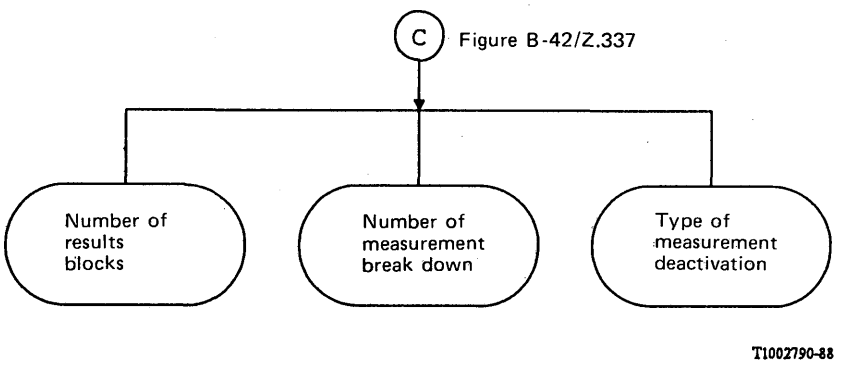


FIGURE B-45/Z.336

Traffic measurement output (continued)

Recommendation Z.337

NETWORK MANAGEMENT ADMINISTRATION

1 General

This Recommendation has been developed in accordance with the methodology defined in Recommendations Z.332 and Z.333.

The main part of this Recommendation deals with the model of the Network Management Administration. A glossary of the terms used is also included.

The list of operator jobs and the list of system functions to be controlled are contained in Annex A.

For each system function to be controlled by means of MML, one or more MML functions can be derived and each of them can be described using the metalanguage defined in Recommendation Z.333, in order to detail the relevant information structure.

Annex B contains guidelines for the definition of the list of MML functions and the information structure diagrams associated to each of them to be used as guidelines.

2 Introduction

Network management is the function of supervising the network and taking action to control the flow of traffic so as to ensure the maximum utilization of the network in all situations. The objective is to enable as many calls as possible to be successfully completed. In its current scope, Network Management doesn't address the aspects of managing traffic on portions of the network that are leased or under the control of network customers.

According to CCITT Recommendations E.410 through E.414, E.502, Q.542 and Q.544, Network Management requires the performance of a certain number of activities to detect abnormal network conditions and to initiate the execution of corrective actions and/or controls. The general scenario of Network Management activities can be described through the involved information flow as depicted in Figure 1/Z.337. Network Raw Data related to traffic parameters and the status (i.e. level of overload, out of service conditions, etc.) of the Network Elements can be processed to provide Network Management Parameters, using the Network Reference Data which are necessary to calculate Network Management Parameters. The Network Raw Data are produced from Network Management Elements.

Network Management Parameters, describing the current network status and performance, can be related to some threshold values (representing the boundary between normal and abnormal behaviour) in order to detect abnormal conditions.

Abnormal Condition Reports, Network Management Parameters and other information (from telephone operators, work centres, news media, etc.) are used to identify the origin of the problem and, as a consequence, to decide the proper actions to be taken or the most suitable controls to be activated.

Network problem identification and consequent corrective actions can be handled manually or automatically by an "expert" system which is capable of performing the functions of abnormal condition detection, problem analysis and problem solving. All the activities carried out for Network Management can be handled by the operator supervision.

Furthermore, for cooperation, coordination and planning purposes, Network Management Reports could be distributed to other operating centres, to higher authorities, etc.

Depending on each Administration and network organization, Network Management activities can be partially or totally performed at exchange level or concentrated in one or more Network Management Centres.

Figure 2/Z.337 contains an example of how the general scenario can be applied to a particular operating organization. In this example, the functions shown for the Network Element and Network Management Operations System are performed by the system under operator control.

Considering the Network Management scenario and the operator activities (listed in Annex A to this Recommendation) three different sub-areas can be recognized:

- Network Management Data administration;
- Network Management Controls administration;
- Network Management Data Distribution administration.

Only the sub-areas for Network Management Data administration is covered by the present Recommendation. The remaining two sub-areas, namely Network Management Controls administration and Network Management Data Distribution administration are left for further study.

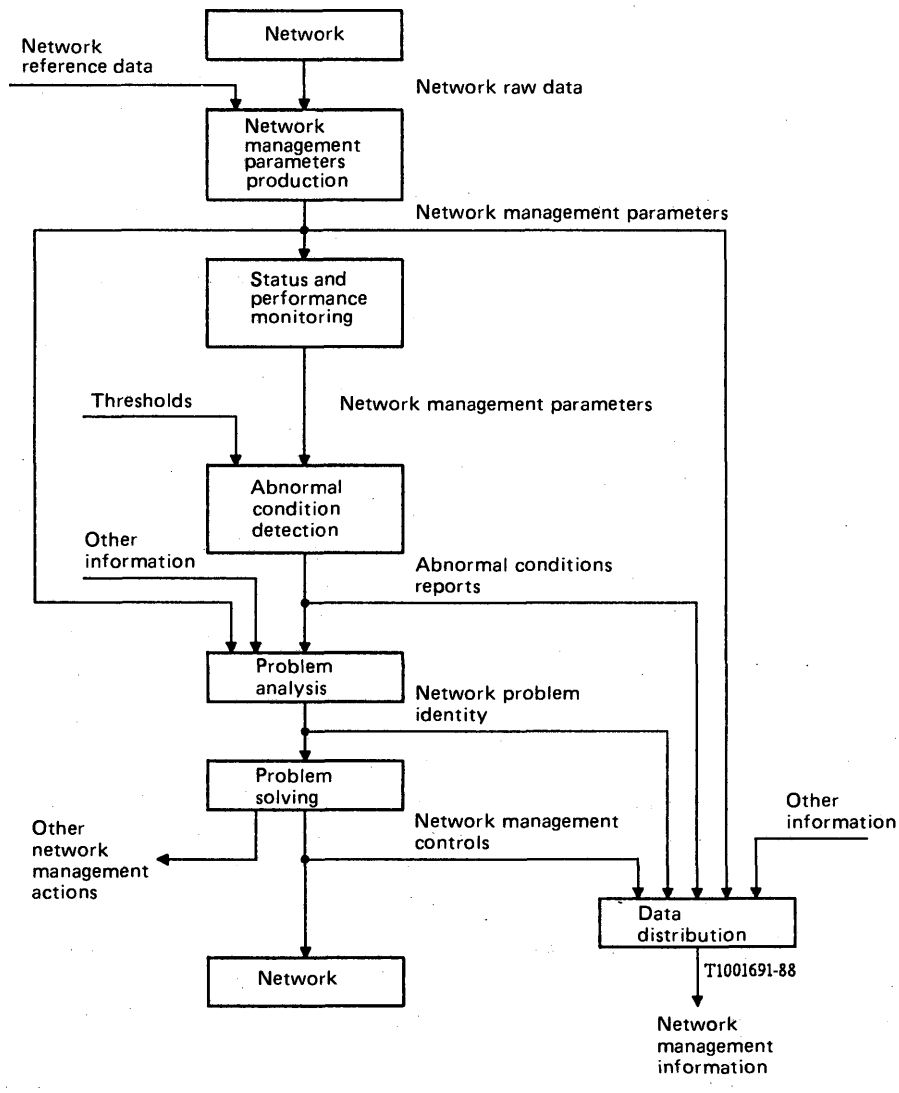
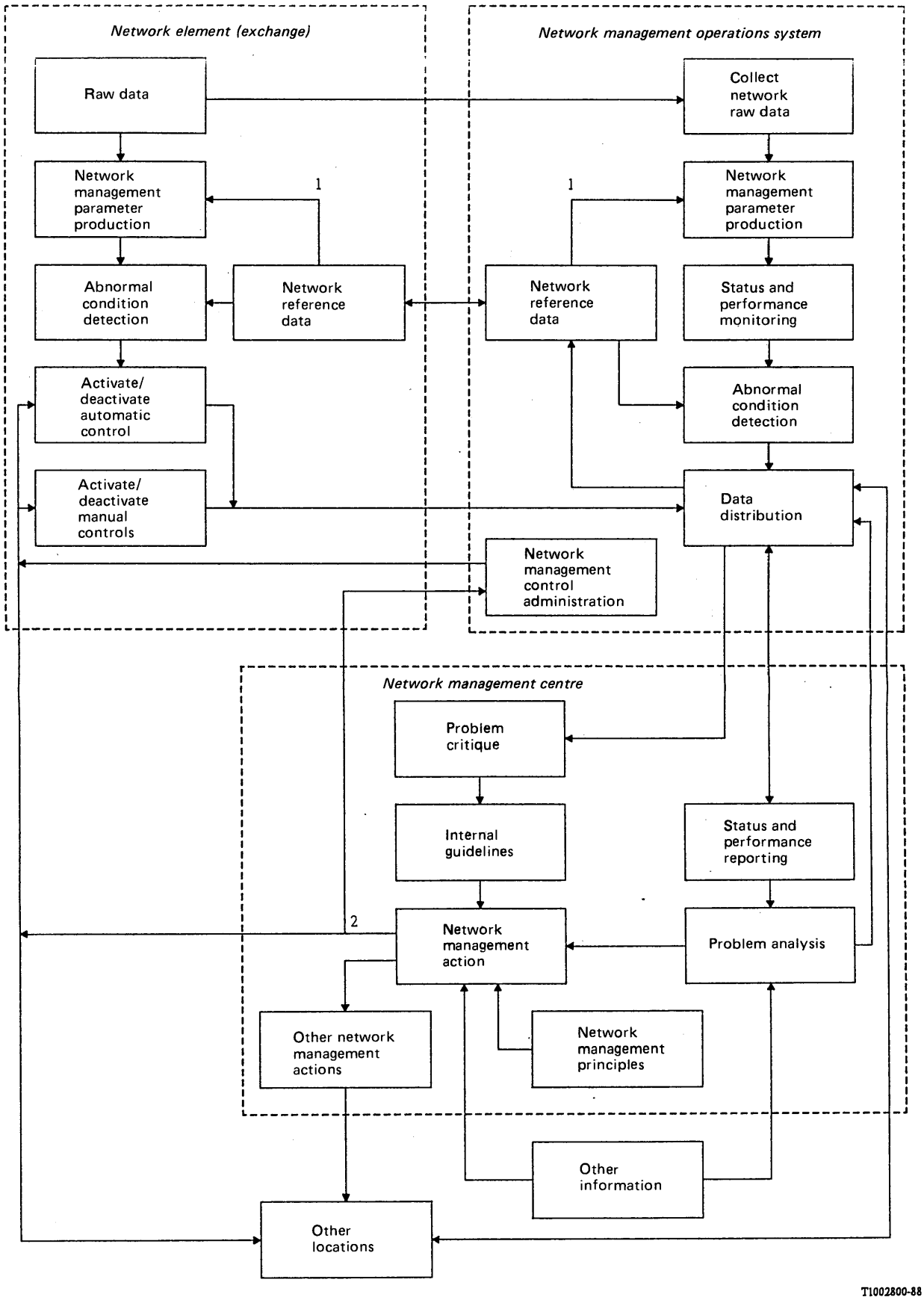


FIGURE 1/Z.337

General network management scenario



Note 1 – Raw data may be processed to produce network management parameters in network elements or in the network management system. In some cases, this function is shared between the network management system and the network element.

Note 2 – Controls may be taken in network elements via the network management system or by direct input to the network element.

FIGURE 2/Z.337

Example of a practical application of the general scenario

3 Network Management Models

3.1 *Introduction*

According to the division of Network Management activities in three sub-areas as stated in § 2, three different models, as indicated in the following, have to be developed:

- Network Management Data administration model;
- Network Management Controls administration model;
- Network Management Data Distribution administration model.

3.2 *Network Management Data administration model*

Network Management Data administration sub-area concerns the operator activities for managing the set of information necessary to monitor the network status and performance. This information can be routed toward proper display and/or storage devices in the Network Management Centre or passed to remote Administrations under operator control.

Data administration is applied to the following data:

- Network Raw Data;
- Network Management Parameters;
- Network Management Indicators;
- Network Reference Data.

3.2.1 *Network Raw Data*

Network Raw Data describe, for each network management element, the current operating conditions (e.g. exchange load, number of in-service circuits, etc.) and traffic behaviour (e.g. number of seizures for circuits sub-group).

Network Raw Data are administered by the MML functions for Traffic Measurements Administration, therefore they are not considered in the Network Management functional area.

3.2.2 *Network Management Parameters*

Network Management Parameters describe the network status and performance in terms of entities (as indicated in Recommendation E.411) and Network Management Objects.

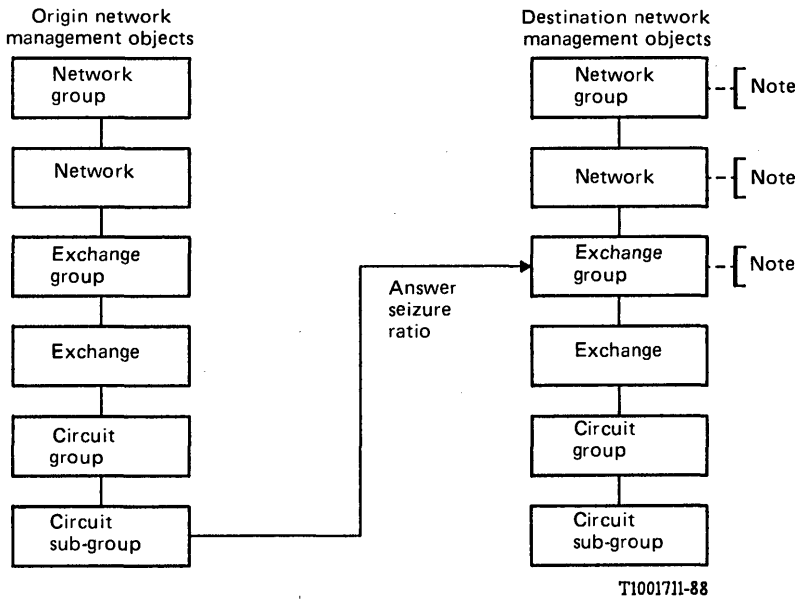
Network Management Objects are the set of Network Elements subject to measurement for Network Management purposes.

One or more Network Management Objects can be grouped as origin and/or destination objects to define traffic flow; the Network Management Objects are hierarchically presented below:

- network group;
- network;
- exchange group;
- exchange;
- circuit group;
- circuit sub-group.

Network management monitoring and control of circuit groups and circuit sub-groups can only be performed via an exchange.

In Figure 3/Z.337 is represented, as an example, a Network Management Parameter in terms of the entity "answer/seizure ratio" for traffic handled by a circuit sub-group and forwarded to a set of exchanges characterized by the same destination code.



Note – Identified only by destination code.

FIGURE 3/Z.337

Example of network management parameter

3.2.3 *Network Management Indicators*

Network Management Indicators represent the logical results (e.g. yes or no) obtained from the comparison of Network Management Parameters against a predefined set of numerical values (thresholds).

Numerical values associated with thresholds can be time-dependent (e.g. morning/afternoon hours, holidays, etc.).

3.2.4 *Network Reference Data*

Network Reference Data can be administered by the MML functions which are appropriate for each type of reference data, therefore they are not considered in the network management functional area.

3.3 *Network management control administration model*

To be developed.

3.4 *Network management data distribution administration model*

To be developed.

4 Glossary of terms

abnormal Condition Report

Information produced in the Network Management Centre after detection of abnormal network status or performance.

circuit group

The set of all switched circuits which directly interconnect one exchange with another.

circuit sub-group

A group of circuits within a circuit group which are uniquely identifiable for operational or technical reasons. A circuit group may consist of one or more circuit sub-groups.

exchange group

A set of exchanges which handles traffic forwarded to or coming from a specific geographical area (e.g. area code, switching centre, etc.).

network

All the exchanges which are relevant to the service standpoint operated by a company in a country.

network Element

Telecommunication equipment which may perform signalling, switching and transmission functions.

network group

A group of telecommunication networks relevant to the service standpoint (e.g. different operating companies offering the same service in the same country).

network management action

The activity performed, not necessarily in the network elements, to regulate traffic flow.

network Management Centre

A centre where network management functions are performed (e.g. O and M centre, switching centre).

network Management Control

The capabilities in network elements to regulate traffic flow and network operation in order to insure the maximum utilization of the network capacity in all situations of traffic overload and network element failure.

network Management Data

The set of information necessary to monitor, detect and identify a network problem.

network Management Indicator

A logical result of comparison of Network Management Parameters and thresholds comparison.

network Management Information

The set of information produced in the Network Management Centre describing the network status and performance, the abnormal conditions detected, the problems identity and the active network management controls.

network Management Object

A set of network elements under control of network management functions and/or subject to measurement for network management purposes.

network Management Parameters

Information produced in the Network Management Centre to be used for the production of abnormal condition report and for display on alerting devices.

network Management System

A system which performs Network Management functions.

network Problem Identity

Information produced in the Network Management Centre to indicate the type of problem detected and the portion of the network and/or services affected.

network Raw Data

Network information provided by network elements and used for the production of Network Management Parameters and for display on alerting devices.

network Reference Data

Information on the network elements and structure (e.g. circuit groups, number of circuits in a circuit group, routing information, type and quantity of switching system components).

ANNEX A

(To Recommendation Z.337)

List of system functions to be controlled by means of MML and list of jobs

A.1 *List of system functions to be controlled by means of MML*

Network Management functions include:

- performing measurements of network status and performance;
- performing Network Management Actions;
- performing Network Management information distribution.

A.2 *List of jobs*

The jobs are supposed to be performed at Network Management Centre level (i.e. any centre which performs network management functions).

A.2.1 *To determine the appropriate Network Raw Data to be collected*

- the purpose of this job is to select the appropriate sets of Network Raw Data for the evaluation of the network elements and traffic monitoring;
- the Network Raw Data are defined for each specific network element. The operator is supposed to select the consistent subset of information necessary for monitoring purposes;
- the complexity of the job is medium;
- the frequency of the job is low.

A.2.2 *To collect appropriate Network Raw Data*

- the purpose of this job is to schedule the collection of Network Raw Data from network elements:
- the operator is supposed to schedule selected sets of Network Raw Data for collection;
- the complexity of the job is medium;
- the frequency of the job is low.

A.2.3 *To define the appropriate parameters to be used for the network and traffic monitoring*

- the purpose of this job is the definition of the Network Management Parameters for the evaluation of the network status and traffic performance. The Network Management Parameters are derived from the available set of Network Raw Data;
- the operator is supposed to define the set of information necessary for Network Management Parameters production;
- the complexity of the job is medium;
- the frequency of the job is low.

A.2.4 *To choose the Network Management Parameters with which to monitor network status and performance*

- the purpose of this job is to choose, from defined Network Management Parameters, a subset for the evaluation of network status and performance;
- the operator is supposed to choose the Network Management Parameters necessary for network monitoring;
- the complexity of the job is medium;
- the frequency of the job is low.

A.2.5 *To activate and deactivate the production of Network Management Parameters*

- the purpose of this job is to control the production of any Specific Network Management Parameter;
- the system is supposed to store the Network Management Parameters produced;
- the complexity of the job is low;
- the frequency of the job is low.

A.2.6 *To assemble appropriate reference data to characterize traffic behaviour of network elements*

- the purpose of this job is to obtain appropriate reference data describing network behaviour to be used in the analysis of the network problems;
- the operator is supposed to determine what reference data is to be collected and stored;
- data items characterizing the network elements and their interrelationships are supposed to be collected and stored in the Network Management Centre;
- the complexity of the job is medium;
- the frequency of the job is low.

A.2.7 *To define or change the thresholds for the network status and performance monitoring*

- the purpose of this job is to define a specific set of thresholds to which Network Management Parameters will be compared;
- the operator is supposed to define, for the selected Network Management Parameters, the numerical values to be used as thresholds;
- the complexity of the job may be medium depending on the number of thresholds to be defined;
- the frequency of the job is low.

A.2.8 *To associate Network Management Parameters with selected thresholds*

- the purpose of this job is to associate selected thresholds with Network Management Parameters;
- the operator is supposed to select the Network Management Parameters which are to be compared against selected thresholds;
- Network Management Parameters are supposed to be compared by the system;
- the complexity of the job is medium;
- the frequency of the job is low.

A.2.9 *To display appropriate exception conditions*

- the purpose of this job is to control the display of the results derived from the comparison of Network Management Parameters against established thresholds so as to alert the operator;
- exception conditions may be displayed on a number of different devices so as to best serve the network management operator. The operator is supposed to select the devices on which exceptions are to be displayed;
- the complexity of the job is medium;
- the frequency of the job is medium.

A.2.10 *To request appropriate additional data displays to characterize network problems*

- the purpose of this job is to request the display of reference data and Network Management Parameters not otherwise automatically displayed. These data elements provide the operator with additional information on which to base his determination of network problems;
- the operator is supposed to determine what additional information is to be displayed;
- the complexity of the job is high;
- the frequency of the job is high.

A.2.11 *To request appropriate data displays to characterize possible network management actions*

- the purpose of this job is to request the display of reference data and Network Management Parameters not otherwise automatically displayed. These data elements provide the operator with additional information on which to base his determination of possible Network Management Actions;
- the operator is supposed to determine what additional information is to be displayed;
- the complexity of the job is high;
- the frequency of the job is high.

A.2.12 *To request additional data analysis for determining actions to be taken*

- the purpose of this job is to request additional data analysis to assist the operator in determining the correct Network Management Actions to be taken (e.g. equipment operating conditions, active Network Management Controls, etc.);
- the operator is supposed to request additional data analysis as to determine alternatives for the implementation of network management controls;
- the complexity of the job is high;
- the frequency of the job is high.

A.2.13 *To choose appropriate Network Management Controls*

- the purpose of this job is to choose the appropriate Network Management Controls to be applied as solutions to network problems;
- the operator is supposed to choose the appropriate controls to correct problems in the network;
- the complexity of the job is high;
- the frequency of the job is high.

A.2.14 *To administer automatic Network Management Controls*

- the purpose of this job is to administer the thresholds/tables used by network elements to implement automatic control in the network;
- the operator is supposed to create, change and delete data from tables used by network elements to implement automatic network management controls;
- the complexity of the job is high;
- the frequency of the job is low.

A.2.15 *To choose appropriate Network Management Control parameters*

- the purpose of this job is to choose the appropriate parameters to be used in the application of network management controls;
- the operator is supposed to select the appropriate control parameters to correct network problems;
- the complexity of the job is high;
- the frequency of the job is high.

A.2.16 *To activate/deactivate Network Management Controls*

- the purpose of this job is to implement Network Management Controls. Implementation may include all types of controls;
- the operator is supposed to implement Network Management Controls;
- the complexity of the job is high;
- the frequency of the job is high.

A.2.17 *To monitor known problems and adjust Network Management Controls*

- the purpose of this job is to provide for iterative adjustment to the network management process. It requires as a minimum that the operator repeat the necessary jobs to insure that actions taken to relieve a particular problem had the optimum effect;
- the operator is supposed to repeat necessary jobs;
- the complexity of the job is high;
- the frequency of the job is high.

A.2.18 *To administer Network Management Information distribution characteristics*

- the purpose of this job is to set up the required parameters for data distribution in and/or outside the Network Management Centre;
- the operator is supposed to define the parameters necessary for display, record and delivery of the Network Management Information;
- the complexity of the job is medium;
- the frequency of the job is low.

A.2.19 *To activate and deactivate Network Management Information Distribution*

- the purpose of this job is to control the distribution of Network Management Information in and/or outside the Network Management Centre;
- the operator is supposed to activate/deactivate automatic and manual Network Management Information Distribution;
- the complexity of the job is medium;
- the frequency of the job is high.

ANNEX B

(To Recommendation Z.337)

Guidelines for the list of MML functions and associated information structure diagrams

B.1 *Introduction*

This annex contains guidelines for the list of MML functions and associated information structure diagrams related to the Network Management administration model defined in Recommendation Z.337 – Section 3.

B.2 *List of MML functions*

The list contains possible MML functions for Network Management administration.

This list is not mandatory nor complete, it may vary according to administration needs, telecommunication network levels, regulatory needs, etc.

These MML functions do not represent the actual command structure of any real implementation of the man-machine interface. Each of the MML functions identified can be implemented by providing one or more separate distinctive commands or several MML functions could be implemented by using a single command.

B.2.1 *List of MML functions for Network Management Data administration*

1) *Creation*
   - create a Network Management Object;
   - create a Network Management Parameter;
   - create a Network Management Indicator.
2) *Activation*
   - activate a Network Management Parameter;
   - activate a Network Management Indicator.
3) *Deactivation*
   - deactivate a Network Management Parameter;
   - deactivate a Network Management Indicator.
4) *Interrogation*
   - interrogate a Network Management Object;
   - interrogate a Network Management Parameter;
   - interrogate a Network Management Indicator.
5) *Deletion*
   - delete a Network Management Object;
   - delete a Network Management Parameter;
   - delete a Network Management Indicator.
6) *Changing*
   - change a Network Management Object;
   - change a Network Management Parameter;
   - change a Network Management Indicator.

B.2.2 *List of MML functions for Network Management Controls administration*

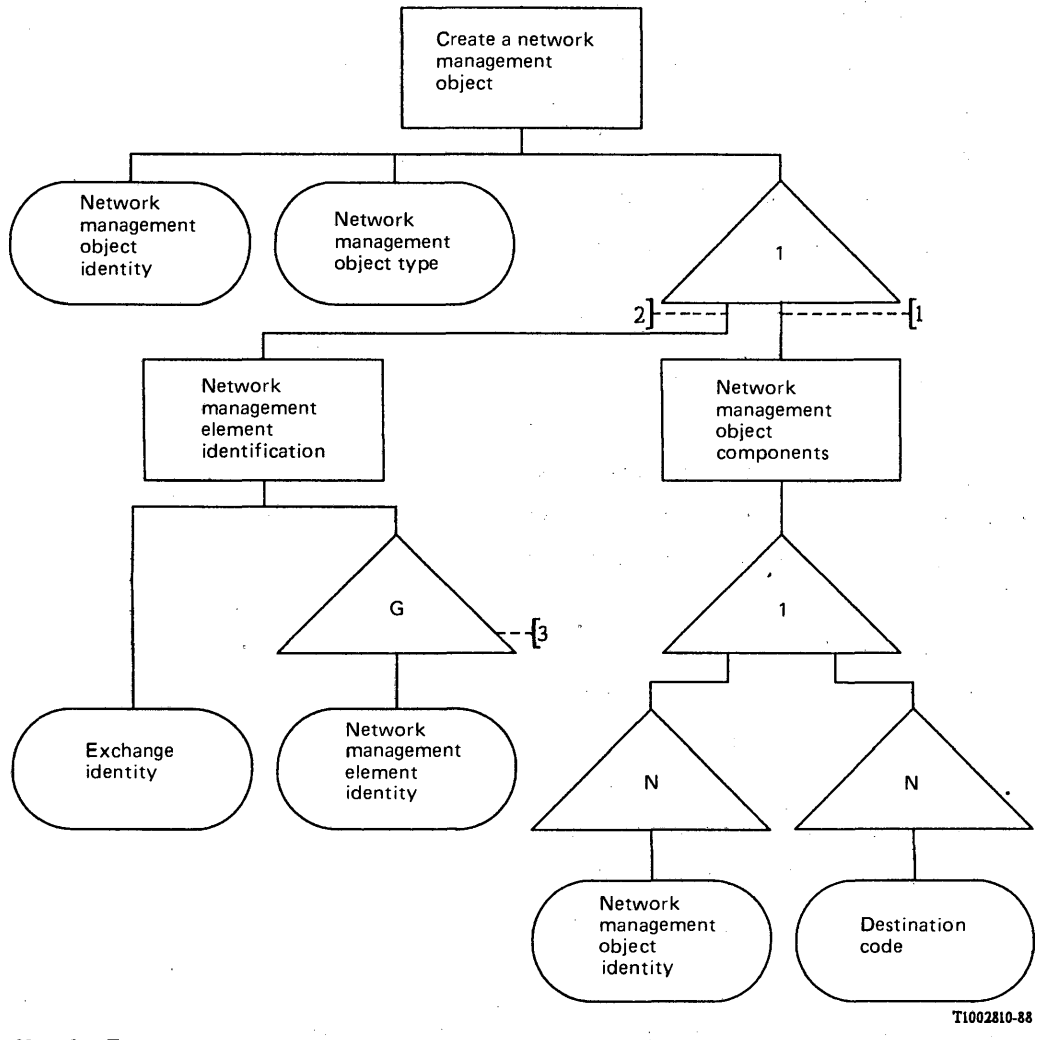
To be developed.

B.2.3 *List of MML functions for Network Management Data Distribution administration*

To be developed.

B.3 *Information structure diagrams*

Only the information entities needed for the MML functions previously defined have been identified and are reported in the present section by means of diagrams representing the information structure of each MML function.



Note 1 – For network management objects constituted as a set of network management objects already identified.

Note 2 – For network management objects directly identified.

Note 3 – Only for circuit sub-group and circuit group identification.

FIGURE B-1/Z.337

Create a network management object

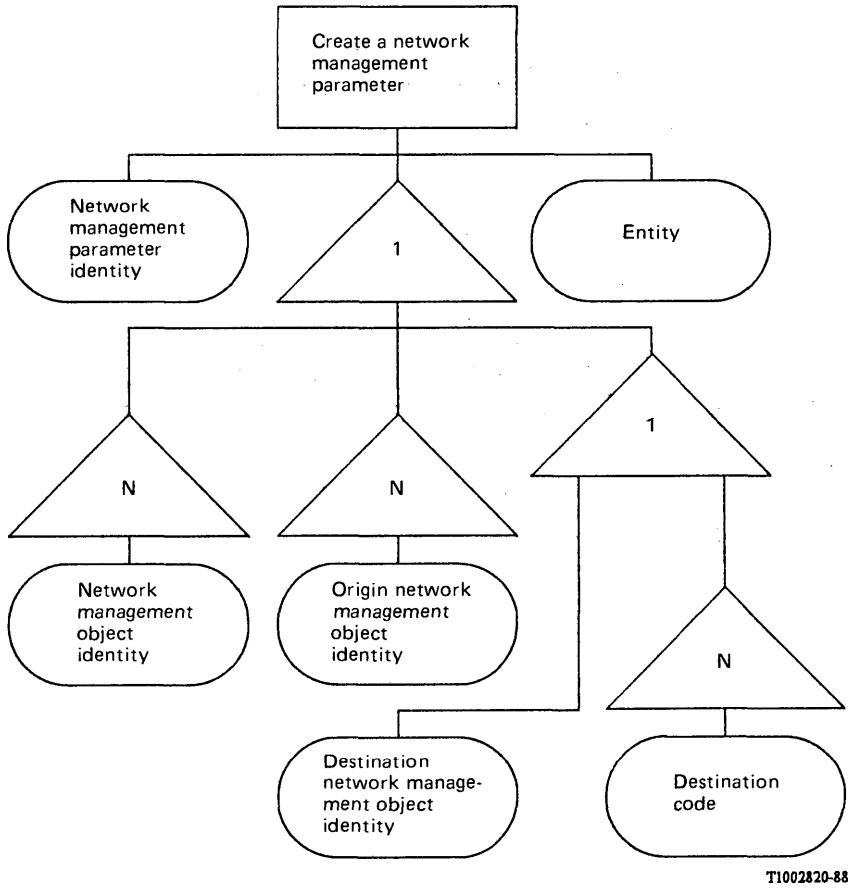


FIGURE B-2/Z.337

Create a network management parameter

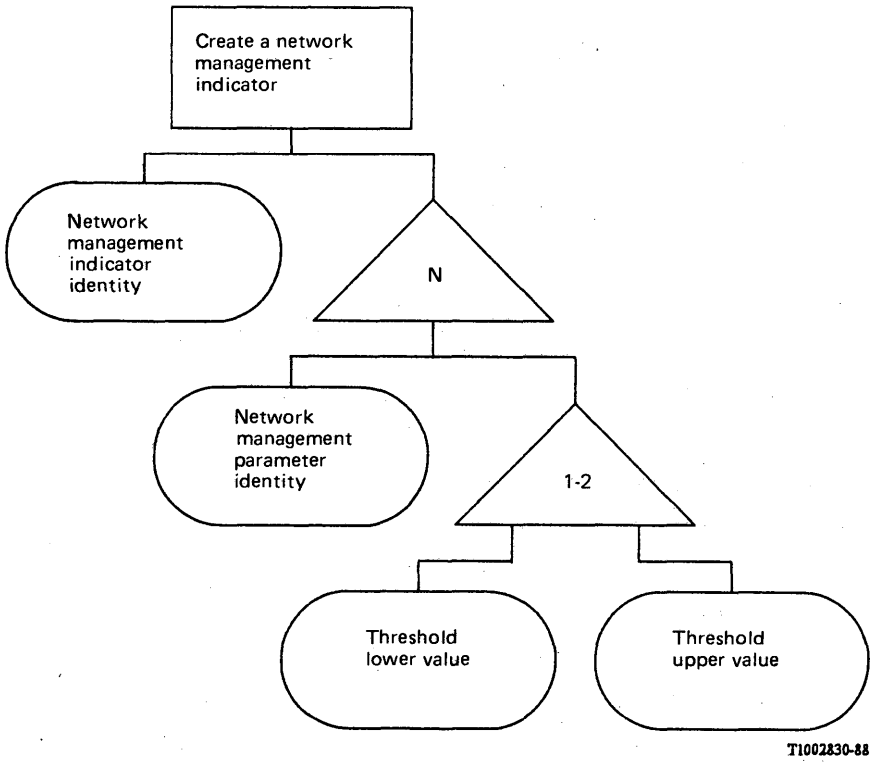


FIGURE B-3/Z.337

Create a network management indicator

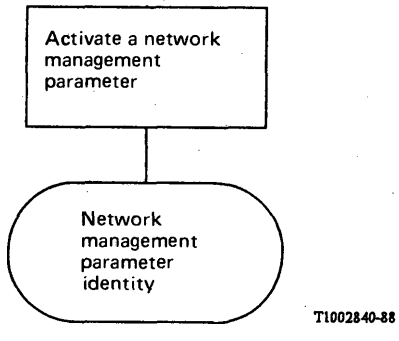


FIGURE B-4/Z.337

Activate a network management parameter

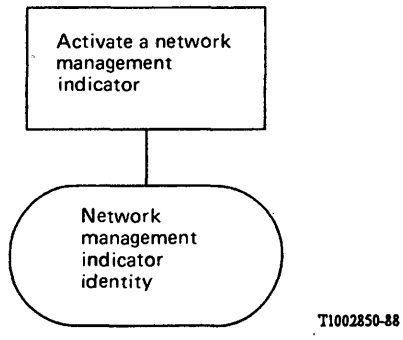


FIGURE B-5/Z.337

Activate a network management indicator

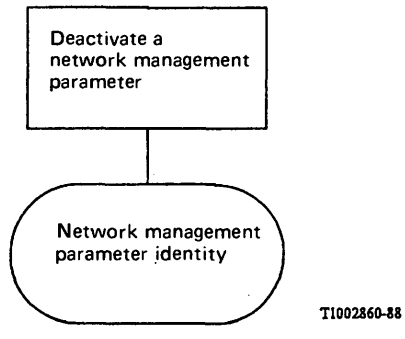


FIGURE B-6/Z.337

Deactivate a network management parameter

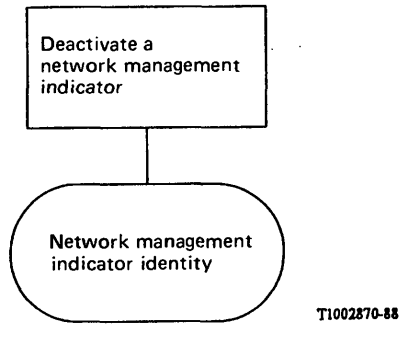
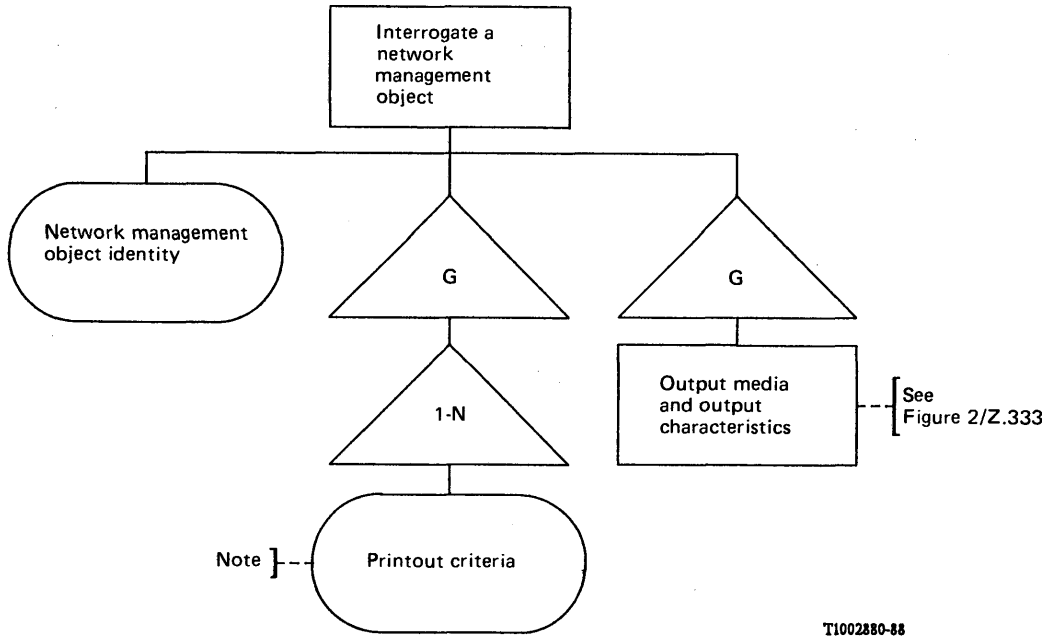


FIGURE B-7/Z.337

Deactivate a network management indicator

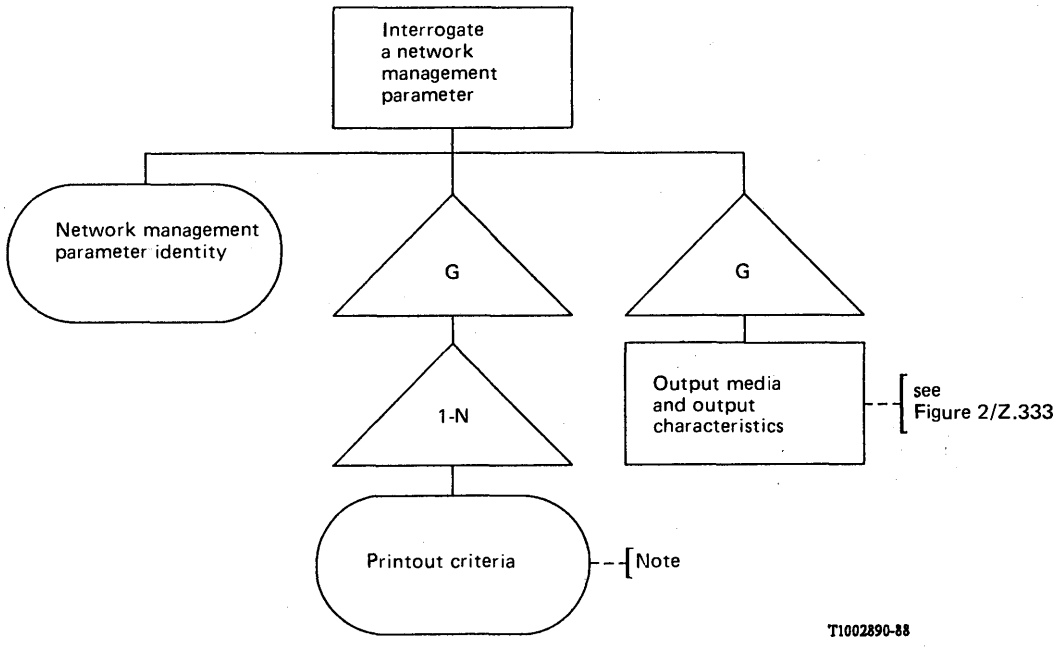


Note – Possible parameter value:

- object type;
- network management object identity;
- associated network management parameters;
- associated network management indicators.

FIGURE B-8/Z.337

Interrogate a network management object

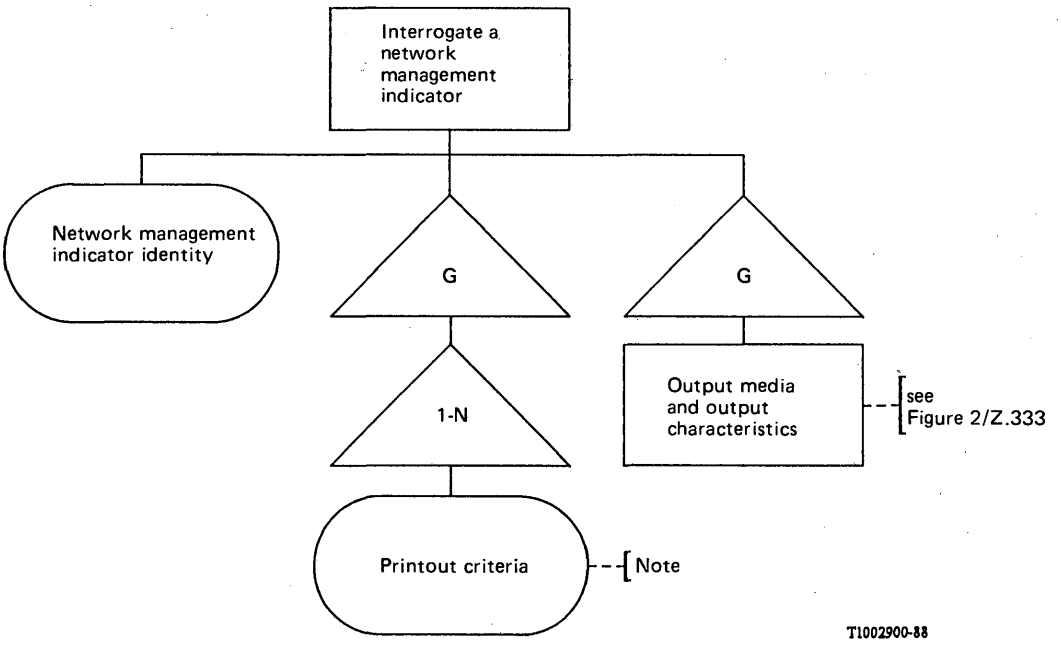


Note – Possible parameter value:

- entity;
- network management object identities;
- destination codes;
- origin network management object identities;
- destination network management object identities;
- associated network management indicators.

FIGURE B-9/Z.337

Interrogate a network management parameter



Note – Possible parameter value:

– network management parameter identity;
– threshold values.

FIGURE B-10/Z.337

Interrogate a network management indicator

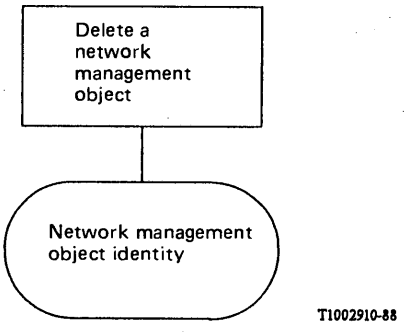


FIGURE B-11/Z.337

Delete a network management object

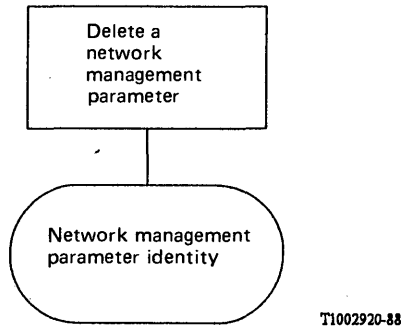


FIGURE B-12/Z.337

Delete a network management parameter

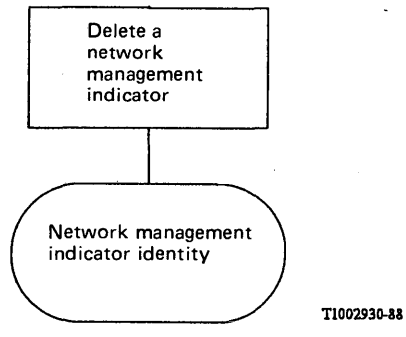


FIGURE B-13/Z.337

Delete a network management indicator

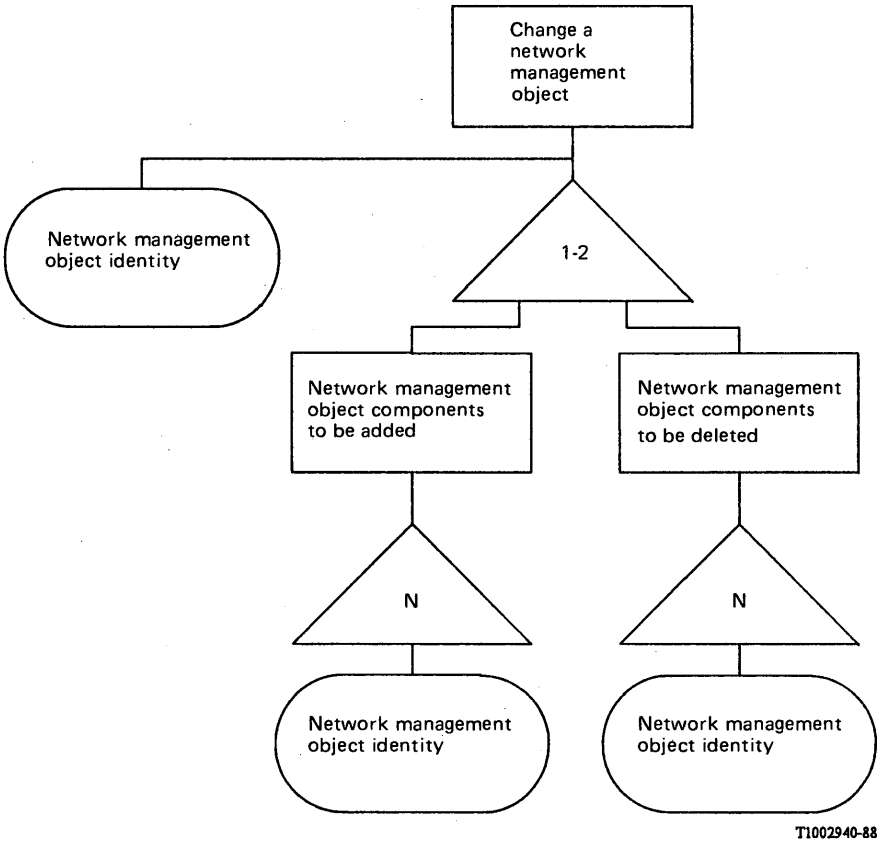


FIGURE B-14/Z.337

Change a network management object

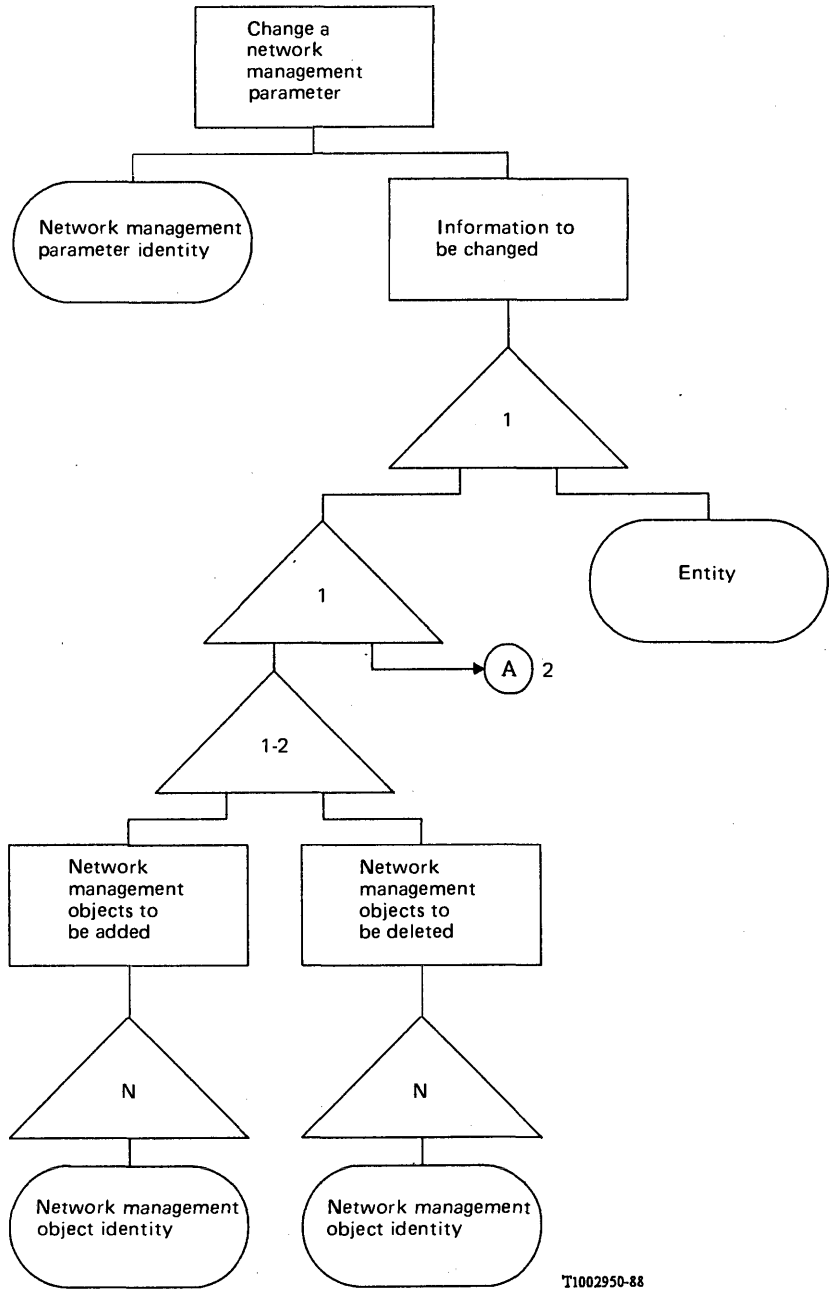


FIGURE B-15/Z.337 (1 of 2)

Change a network management parameter

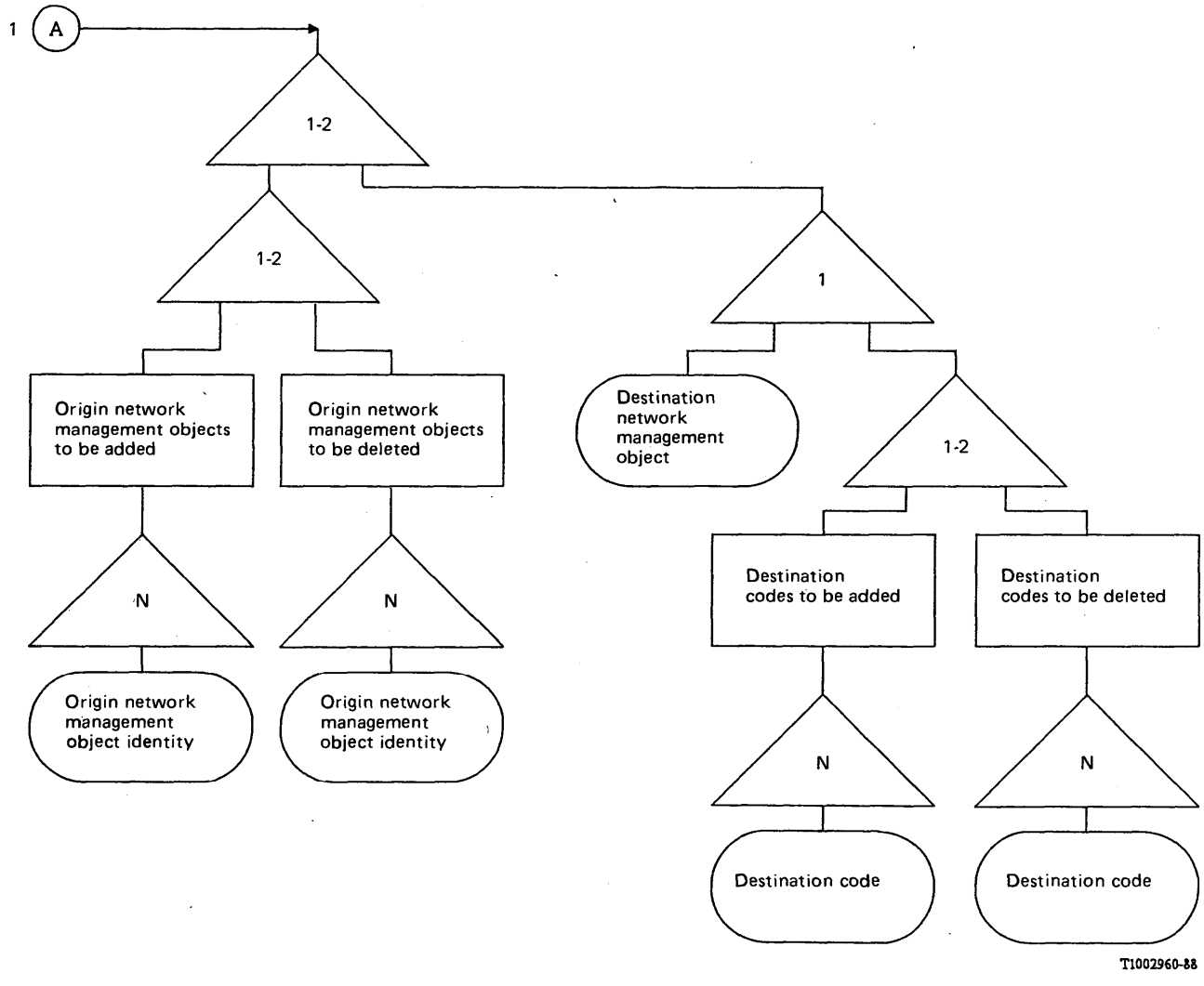


FIGURE B-15/Z.337 (2 of 2)

Change a network management parameter

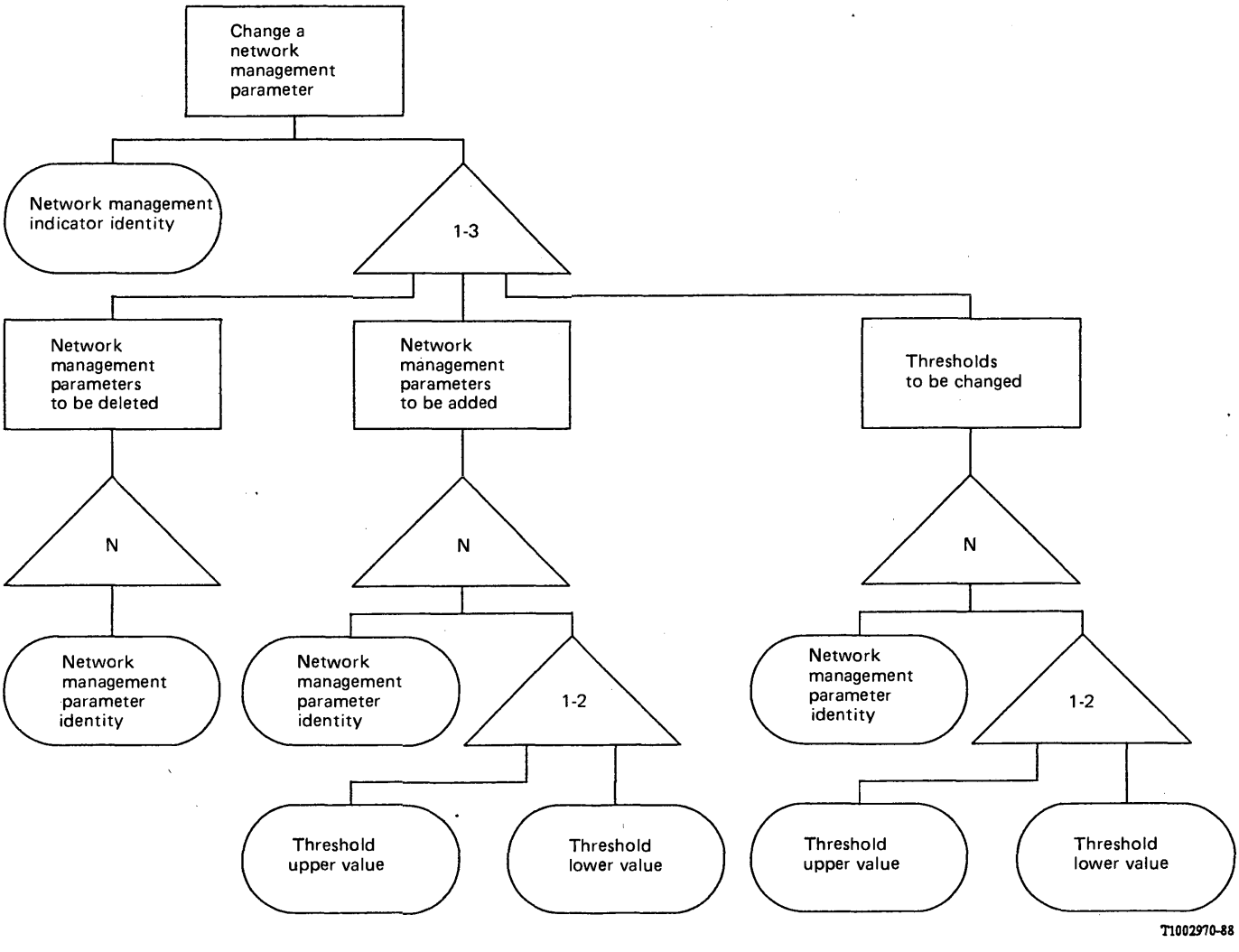


FIGURE B-16/Z.337

Change a network management indicator

GLOSSARY OF TERMS

1 General

The aim of the glossary for the man-machine language is to include terms used in describing the man-machine language. It comprises, in alphabetical order, terms used in the Z.300-series Recommendations that have a special significance in the MML context and hence require definition. Terms comprising words used in their ordinary, every day sense, i.e. unambiguous and self-explanatory words, are not included.

The terms in italics in the text of the definitions are defined elsewhere in this glossary. If a term has one meaning within the context of Recommendations Z.321-Z.323, and another within the context of Recommendations Z.331-Z.333, then the meaning in the former context follows i), and the meaning in the latter follows ii).

2 List of terms

abnormal condition report

F: rapport de condition anormale

S: informe de condición anormal

Information produced in the *network management centre* after detection of abnormal *network* status or performance.

acceptance input

F: entrée d'acceptation

S: entrada de aceptación

An *input* used to allow the *system* to *output* a high priority message, announced by a *message waiting indication.*

acceptance output

F: sortie d'acceptation

S: salida de aceptación

An *output* message indicating that an *input* to the *system* is syntactically correct and complete and that the appropriate *system actions* will be initiated, or have already been carried out. In the latter case, this indication may take the form of the actual result.

accessible field

F: champ accessible

S: campo accesible

A *field* for writing by the *user* and the *system.*

action

F: action

S: acción

The process of performing an *MML function* ; usually represented by a verb.

action modifier

F: modificateur d'action

S: modificador de acción

A qualification of an *action.*

activate

F: activer

S: activar

An *action* to initiate a *system* process that requires preliminary data entry, or an *action* to make previously entered *data set* available to the *system* for its intended use; opposite of *deactivate.*

additional header information

F: information supplémentaire d'en-tête

S: información adicional de encabezamiento

Provides information supplementary to the actual *output header*, such as sequence number, processor number, *output* device, or day of the week.

additional information

F: information supplémentaire

S: información adicional

i) General information on how to proceed, e.g. how to select an item, a *form*, a *menu* or how to submit a *form* to the *system.*

ii) List of possible values to be associated with one or more *information entities* in *information structure diagrams.*

administrative system

F: système d'administration

S: sistema administrativo

A *system* which supports administration personnel in performing administrative *jobs*, e.g. billing, related to *SPC systems.*

alarm statement

F: instruction d'alarme

S: sentencia de alarma·

A statement providing information concerning an alarm condition, such as the degree (level) of alarm or the source of the alarm.

allow

F: autorisation

S: permitir

An *action* to permit specified *system actions*, responses, or *functions* to occur; these *functions* may be inhibited by *system* design or by use of the *inhibit action.*

annotation

F: annotation

S: anotación

An aspect of the *drawing convention* of the *syntax and decomposition meta-language* indicating how descriptive or explanatory notes may be presented for clarification purposes.

annotation symbol

F: symbole d'annotation

S: símbolo de anotación

A *symbol* (– – – – –[n where *n* is a number referencing a note) used in the syntax *meta-language* for *annotation* purposes.

application

F: application

S: aplicación

A set of *functions* required to perform a *job.*

arithmetic delimiter

F: délimiteur arithmétique

S: delimitador aritmético

A *symbol* used to delimit an *arithmetical expression*: ((left parenthesis) for the opening *delimiter* and) (right parenthesis) for the closing *delimiter.*

arithmetic operator

F: opérateur arithmétique

S: operador aritmético

A *symbol* used to denote the arithmetic operation(s) to be performed in an *arithmetical expression.* Allowed operators are: + (plus sign), – (hyphen), / (solidus), * (asterisk).

arithmetical expression

F: expression arithmétique

S: expresión aritmética

A combination of *arithmetic operators, numerals* (*decimal, hexadecimal, octal* or *binary*) and *identifiers* enclosed by *arithmetic delimiters.*

auxiliary system

F: système auxiliaire

S: sistema auxiliar

A *system* that supports *SPC systems* in performing their tasks. It may be either an *operation and maintenance system* or an *administrative system.*

Backus Naur Form (BNF)

F: forme de Backus Naur (FBN)

S: forma Backus Naur (FBN)

A syntactic *meta-language* for use in specifying the syntax structure of *inputs* and *outputs* of an actual *man-machine interface.*

binary numeral

F: numéral binaire

S: numeral binario

A *numeral* in the binary (base 2) *numbering system,* represented by the *characters* 0 (zero), 1 (one) and optionally preceded by B' (B apostrophe).

block mode transmission

F: transmission en mode bloc

S: transmisión en modo bloque

A transmission characteristic in which all of the regular typewriter keys and some of the special purpose keys are only transmitted to the controlling processor, in a block, when a "send" key is activated.

block of parameters

F: bloc de paramètres

S: bloque de parámetro

A set of *parameters* containing information necessary for the *system* to perform the *function* specified in the *command.*

border area

F: zone périphérique

S: zona de marco (ó lateral)

That part of a *visible display* which is physically unavailable for displaying or entering data.

browse

F: lecture

S: hojear

An *action* to display sequentially the current values of items in a *data set*; the *user* may examine the data items in either the forward or backward direction.

CCITT MML

F: LHM du CCITT

S: LHM del CCITT

The *man-machine language* (MML) developed by the International Telegraph and Telephone Consultative Committee (CCITT) for *stored program-controlled systems* and *operation and maintenance systems.*

change

F: modifier

S: cambiar

An *action* to modify specified data items in a *data set.*

character mode transmission

F: transmission en mode caractère

S: transmisión en modo carácter

A transmission characteristic in which each and every character *input* at the keyboard is sent to the controlling processor one at a time.

character set

F: jeu de caractères

S: juego de caracteres

The finite set of different characters used in *CCITT MML.*

circuit

F: circuit

S: circuito

Connection between two *exchanges* for one call at a time, including the junctors that terminate the circuit.

circuit group

F: faisceau de circuits

S: haz de circuitos

The set of all switched *circuits* which directly interconnect one exchange with another.

circuit subgroup

F: sous-faisceau de circuits

S: subhaz de circuitos

A group of *circuits* within a *circuit group* which are uniquely identifiable for operational or technical reasons. A *circuit group* may consist of one or more *circuit subgroups.*

circuit subgroup

F: sous-groupe de circuits

S: subhaz de circuitos

Group of *circuits* between two *exchanges* having the same traffic direction (incoming, outgoing, bidirectional), the same signalling characteristics and the same transmission medium characteristics.

clarifying text

F: texte explicatif

S: texto aclaratorio

A set of *information units* used to make the purpose and content of the *output* clearer.

class A function

F: fonction de la classe A

S: función de clase A

A *function* which provides the user with the means to control *system functions* via MML *inputs* and *outputs*; also known as an *MML function.* It can be viewed as an *action* upon an *object.*

class B function

F: fonction de la classe B

S: función de clase B

A *function* which can be controlled at least partially by the *user* by means of *class A* (or *MML*) *functions.*

class C function

F: fonction de la classe C

S: función de clase C

A *function* which is not controllable by the *user* in a given *system.*

command

F: commande

S: instrucción

The complete specification of a *function* that the *system* is required to perform. It comprises a *command code* followed generally (but not necessarily) by one or more *blocks of parameters.*

command code

F: code de commande

S: código de instrucción

A set of up to 3 *identifiers,* each separated by a - (hyphen), used to define the nature of the *command.*

command entry sequence

F: séquence d'introduction de commande

S: secuencia de introducción de instrucción

The sequence of operations required to input a *command* or a series of *commands.*

command reference

F: référence de commande

S: referencia de instrucción

A reference to a previously given *command,* appearing in *output outside dialogue* and *dialogue procedures,* in the form of a *command sequence number* and, possibly, *clarifying text.*

command sequence number

F: numéro de séquence de commande

S: número secuencial de instrucción

A reference number uniquely identifying a *command* recognized by the *system.*

comment

F: commentaire

S: comentario

A character string enclosed between the *separators* /* (solidus asterisk) and */ (asterisk solidus). It has no MML syntactical or semantical meaning.

component

F: composant

S: componente

A *decomposition meta-language* symbol for an *information entity* that cannot be divided further.

composite part

F: partie composite

S: parte compuesta

A *decomposition meta-language* symbol for an *information entity* that can be divided into smaller parts.

compound parameter argument

F: argument de caractère composé

S: argumento de parámetro compuesto

A *parameter argument* made up of more than one *information unit.* It is used to specify a multidimensional *object* or value, e.g. a date can be expressed as 1979-12-31.

concealment

F: masquage

S: ocultación

A *video attribute* by which information is hidden, e.g. secret parts of a password.

condition

F: condition

S: condición

An *identifier* and a (group of) *parameter argument(s)* separated by a *relational operator.* Used in data base queries.

connectivity rules

F: règles de connectivité

S: reglas de conectividad

An aspect of the *drawing convention* of the *decomposition meta-language* indicating *symbol* interrelationship.

connector

F: connecteur

S: conector

An aspect of the *drawing convention* of the *decomposition meta-language* indicating how *flowlines* may be broken.

continuation character

F: caractère suite

S: carácter de continuación

A special *execution character* implying a similar *command code* for the next *command* and hence allowing the *system* to *prompt* directly for the next *block of parameters.*

control character

F: caractère de commande

S: carácter de control

A character whose occurrence in a particular context initiates, modifies, or stops an *action* that affects the recording, processing or interpretation of data.

control functions

F: fonctions de commande

S: funciones de control

Functions related to the *man-machine interface* that are applied by the *user* independently while in a *dialogue* with the *system* application *functions. Control functions* have no direct impact on the *system functions.*

control key

F: touche de commande

S: tecla de control

A key which when pressed performs a *control function.*

correction character

F: caractère de correction

S: carácter de corrección

A character used to invoke correction facilities prior to analysis of *input* by the *system.*

cursor control functions

F: fonctions de commande de curseur

S: funciones de control de cursor

Functions influencing the position or movement of the *cursor.*

create

F: créer

S: crear

An *action* to establish in the *system* a new *data set*; opposite of *delete.*

cursor

F: curseur

S: cursor

The item in the *display area* which identifies the position appropriate to the task at hand, e.g. where the next character will appear.

data set

F: ensemble de données

S: conjunto de datos

A user-accessible set of one or more data items characterized by a particular use and also by the constraints on data format and/or values that make it suitable for this use.

deactivate

F: désactiver

S: desactivar

An *action* to terminate a *system* process initiated by an *activate action*, or an *action* to make a *data set* unavailable for use by the *system*; opposite of *activate.*

decimal numeral

F: numéral décimal

S: numeral decimal

A *numeral* in the decimal (base 10) *numbering system*, represented by the characters 0 (zero), 1, 2, 3, 4, 5, 6, 7, 8, 9 optionally preceded by D' (D apostrophe).

decomposition meta-language

F: métalangage de subdivision

S: metalenguaje de descomposición

A graphical *meta-language* to describe the structure of the *information entities* associated with an *MML function.*

default option

F: option par défaut

S: opción por defecto

A *symbol* of the *decomposition meta-language* which indicates that the value taken by an *information entity* will be provided automatically if the user does not supply a value in the *input* for such an *information entity.*

default value

F: valeur par défaut

S: valor por defecto

The value given to any *parameter* by the *system* in the absence of a specific value in the *user's input.*

delete

F: suppression

S: borrar

An *action* to eliminate a *data set* from the *system*; opposite of *create.*

delimiter

F: délimiteur

S: delimitador

A character that organizes and separates items of data.

destination identifier

F: identificateur de destination

S: identificador de destino

Identifies, after *input*, the system (destination) that, from the *user*'s perspective, becomes the new partner in a *dialogue.*

destination prologue

F: prologue de destination

S: prólogo de destino

An operating sequence causing subsequent *inputs* to be processed in the *system* defined by the *destination identifier.*

dialogue

F: dialogue

S: diálogo

See *dialogue procedure.*

dialogue element

F: élément de dialogue

S: elemento de diálogo

Element of a set of three types of *information entry* in a *man-machine communication*: viz. *direct information entry, information entry* through *menu-item selection* or through *form filling.*

dialogue procedure

F: procédure de dialogue

S: procedimiento de diálogo

The complete interactive procedure for interchanging data between *user* and *system* comprising *procedure prologue, procedure body* and *procedure epilogue.* In the Z.300-series Recommendations, the terms *dialogue* and *dialogue procedure* are interchangeable.

digit

F: chiffre

S: cifra; digito

A character of the *character set* representing an integer listed in Table 1/Z.314, column 3, positions 0 (zero) to 9.

direct information entry

F: introduction directe d'information

S: introducción directa de información

A *dialogue element* whereby the *input* of a *command* or *destination identifier* is done without the aid of *menus* and/or *forms.*

directive

F: directive

S: directriz

Input to direct the *system* to present information rather than to execute a *command*; can also be used in the interaction between the *user* and *system* prior to *command* execution. Directives can never cause any change in the state of the *system.*

display area

F: zone de visualisation

S: zona de visualización

That part of a *visible display* which is available for displaying or entering data.

displayed form

F: formulaire affiché

S: formulario visualizado

A *form* filled out and displayed by the *system* upon request by the *user.*

documents A through G

F: documents A à G

S: documentos A a G

Specially formatted information generated during various *phases* of the *methodology* for the specification of the *man-machine interface.*

drawing convention

F: convention de tracé

S: convenio de representación

A set of rules provided by the *decomposition meta-language* to indicate the allowed use of the *symbols* and their interconnection.

edit

F: éditer

S: editar

An *action* to display a specified *data set* and subsequently to modify the *data set.*

end of dialogue

F: fin de dialogue

S: fin de diálogo

The indication that *dialogue* has finished.

end of input indication

F: fin d'indication d'entrée

S: fin de indicación de entrada

An indication to mark the end of *input* in order to have the information interpreted by the *system.*

end of output

F: fin de sortie

S: fin de salida

The indication that *output outside dialogue* has finished.

end statement

F: instruction de fin

S: sentencia de fin

Terminates *output* information from the *system* in an operating sequence where termination is not obvious.

error correction

F: correction d'erreur

S: corrección de error

The activity of correcting *input* which has been offered to but not accepted by the *system.*

escape indication

F: indication d'échappement

S: indicación de escape

A mechanism to indicate that following character(s) are not to be interpreted according to the normal *syntax* rules.

exchange

F: central

S: central

SPC switching *system.*

exchange group

F: groupe de centraux

S: grupo de centrales

A set of *exchanges* which handles traffic forwarded to or coming from a specific geographic area (e.g. area code, switching centre, etc.).

execution character

F: caractère d'exécution

S: carácter de ejecución

A character which requests that the *command* be executed.

field

F: champ

S: campo

A part of a *window* area (sometimes the entire *window* area), which is used either for entering or displaying information.

filter

F: filtre

S: filtro

An *action* to form a subset of a *data set* consisting of all data items in the *data set* meeting specified criteria; the original *data set* is unaffected by this *action.*

flowline

F: ligne de liaison

S: línea de flujo

A line representing a connection between *symbols* in:

i) a *syntax diagram*;

ii) an *information structure diagram.*

form

F: formulaire

S: formulario

A list of *parameters,* including empty positions for insertion of *parameter values* by the *user.*

form filling

F: remplissage de formulaire

S: cumplimentación de formulario

The activity of inserting *parameter values* into a *form,* and submitting the completed *form* to the *system* under *user* control.

form identity

F: identité de formulaire

S: identidad de formulario

An identity unique to a *form* so that it can be distinguished from other *forms.*

form output

F: sortie de formulaire

S: salida de formulario

An *output* of a *form* belonging to a *command,* used in certain *information entry* procedures.

format effector

F: caractère de mise en page

S: determinante de formato

Any character(s) used to control the position of printed, displayed or recorded data.

function

F: fonction

S: función

A *system* activity necessary to the performance of a duty for which the *system* was designed (see also *class A, B, and C functions*).

functional area (or sub-area)

F: domaine fonctionnel (ou sous-domaine)

S: área (o sub área) funcional

A set of related operation, maintenance, installation or acceptance testing *functions* to be controlled by means of *MML* (*class B functions*).

function key

F: touche de fonction

S: tecla de función

A key which when pressed causes a modification in the *man-machine terminal* or causes the *system* to perform a specific *function.*

function model

F: modèle de fonction

S: modelo de función

A formal or informal representation of one or more aspects of those parts of telecommunication systems which should be controlled by means of MML.

general option

F: option générale

S: opción general

A *symbol* of the *decomposition meta-language* which indicates either that an *information entity* exists in the *system* in a predetermined manner or that it is not needed.

general information window area

F: sous-fenêtre d'information générale

S: zone de ventana de información general

This *window area* can contain system identification and/or application identification, date, time, and other relevant information.

graphic characters

F: caractères graphiques

S: caracteres gráficos

A collection of characters within the *character set* used to improve readability of *output.*

graphic terminals

F: terminaux graphiques

S: terminales gráficos

Terminals which provide graphic capability (line drawing, circles, etc.) using other than alphanumeric means.

guidance output

F: sortie de guidage

S: salida de orientación

Output providing assistance to the *user* in a *man-machine* communication.

guidelines

F: lignes directrices

S: líneas directrices

i) Information that gives general direction in the implementation of *CCITT MML.*

ii) General directions by which the purpose of one or more phases of the *methodology* may be accomplished.

header

F: en-tête

S: encabezamiento

General information which could comprise identification information, date and time, etc.

help request

F: demande d'assistance

S: petición de ayuda

User input to ask for assistance.

help output

F: sortie d'assistance

S: salida de ayuda

The *output* resulting from a request for *user* assistance.

hexadecimal numeral

F: numéral hexadécimal

S: numeral hexadecimal

A *numeral* in the hexadecimal (base 16) *numbering system*, represented by the characters 0 (zero), 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, optionally preceded by H' (H apostrophe).

highlighting

F: renforcement

S: resaltación

Techniques used to emphasize visually a portion of the *display area* to make it stand out from adjacent portions, i.e. to call the viewer's attention to it.

identification invitation

F: invitation à identification

S: invitación a la identificación

A prompt to request the *user* to identify himself by means of a *password* and/or an identity card.

identifier

F: identificateur

S: identificador

A representation of an entity, typically consisting of one or more *characters*. It is used to identify or name a unique item of data. In the *man-machine language*, the first character is a letter.

inaccessible field

F: champ inaccessible

S: campo inaccesible

A field for writing only by the *system*.

index number

F: indice

S: número índice

A character combination consisting of one or more *digits*. Used in compound *parameter names*.

indicator

F: indicateur

S: indicador

A *character input* by a *user* or *output* by a *system* to indicate a state or to request *user* or *system* action.

information entity

F: entité d'information

S: entidad de información

An information element associated with an *MML function* and usually represented in an *information structure diagram.*

information entry

F: introduction d'information

S: introducción de información

General term for each of the three *dialogue elements.*

information entry through form filling

F: introduction d'information par remplissage de formulaire

S: introducción de información por cumplimentación de formulario

A *dialogue element* whereby the *input* of *parameter values* is done by means of *form filling.*

information entry through menu-item selection

F: introduction d'information par sélection en mode menu

S: introducción de información por selección de elemento de menú

A *dialogue element* whereby the *input* of a *command* or *destination identifier* is done by means of *menu-item selection.*

information structure (diagram)

F: structure d'information (diagramme)

S: estructura de información (diagrama de)

A representation of the *information entities* associated with an *MML function* and their interrelationships.

information structure meta-language

F: métalangage de structure d'information

S: metalenguaje de estructura de información

See *decomposition meta-language.*

information unit

F: unité d'information

S: unidad de información

The smallest part of data in the *input* or *output.*

inhibit

F: interdire

S: inhibir

An *action* to prevent the specified *system actions*, *system* responses or *functions* from occurring; these *functions* may normally be allowed by the *system* design or by the *allow action.*

initialize

F: initialiser

S: inicializar

An *action* to put specified data or equipment into a predefined initial (normal) condition or value.

input

F: entrée

S: entrada; introducir

i) Information that is transferred to the *system* by the *user*, e.g. *commands*, *directives*, *menu-item selections*, *form identities*, etc.

ii) An *action* to enter data by means of a *man-machine terminal* into the *system.*

input acknowledgement

F: accusé de réception d'entrée

S: acuse de entrada

Termination of *information entry* through *menu-item selection* or *form filling.*

input error

F: erreur d'entrée

S: error de entrada

A *system*-detected error in *input* information.

input error information

F: information d'erreur d'entrée

S: información de error de entrada

Information describing the location and nature of an *input error.*

input field

F: champ d'entrée

S: campo de entrada

See *accessible field.*

input window area

F: sous-fenêtre d'entrée

S: zona de ventana de entrada

See *output* and *input window area.*

interaction request output

F: sortie de demande interactive

S: salida de petición de interacción

System output inviting further *user actions.*

interactive

F: interactive

S: interactiva

A condition where *information entry* can be done by the *user.*

interactive operating sequence

F: séquence d'exploitation interactive

S: secuencia operativa interactiva

A *sequence* which may consist of a single *command entry sequence* terminated by an optional *end statement* or of a series of command entry *sequences* and/or *manual responses.* The latter occurs when, as a result of partial execution of a *function,* the *system* requests the *user* to supply it with further information in the form of *manual responses* or further *commands* for which *user* judgement and/or decision is required.

interface control functions

F: fonctions de commande d'interface

S: funciones de control de interfaz

Functions used to force specific *actions* relating to the interface.

interrogate

F: interroger

S: interrogar

An *action* to provide a display of the current value of the items of one or more *data sets.*

inverse video

F: inversion vidéo

S: inversión video

A *video attribute* by which information can be displayed by inverting the image of the characters, such as going from light characters on a dark background to dark characters on a light background.

item description

F: description de rubrique

S: descripción de elemento

A brief description of the nature of the item in a *menu.*

item selection procedure

F: procédure de sélection de rubrique

S: procedimiento de selección de elemento

A procedure to select an item out of a list of items on a *menu output.*

iteration

F: itération

S: iteración

A *symbol* of the *decomposition meta-language* which indicates that a repetitive use of one or more *information entities* is possible.

I/O device

F: dispositif d'E/S

S: dispositivo de E/S

Device for entering or receiving data to or from a *system.* Can be controlled manually for entering or receiving data.

job

F: tâche

S: trabajo

A discrete administrative activity within a telecommunications business which is designated as a part of the overall plan for running the business and characterized by *man-machine communication.*

job area

F: domaine de tâches

S: aréa de trabajo

A collection of jobs particular to a given *functional area,* e.g. subscriber line maintenance, trunk line maintenance, call routing administration, etc.

key parameter

F: paramètre clé

S: parametro clave

A term used in data base techniques to uniquely identify a data record.

keyed numeral

F: numéral clavier

S: numeral de teclado

A *numeral* in a *numbering system* based on keypad *input,* represented by the characters 0 (zero), 1, 2, 3, 4, 5, 6, 7, 8, 9, *, #, A, B, C, D, optionally preceded by K' (K apostrophe).

layout option

F: option de présentation

S: opción de estructuración (de la presentación)

A combination of *format effectors* and/or *graphic characters* used to bound elements of the *output* in a clear and readable form.

letter

F: lettre

S: letra

A character of the *character set* representing the alphabet, listed in Table 1/Z.314, columns 4, 5, 6 and 7 excluding table positions 5/15 and 7/15.

line group

F: groupe de lignes (ligne groupée)

S: grupo de líneas

A line group is a group of lines of a multi line subscriber with some common line characteristics, e.g. incoming, outgoing, bothway.

machine

F: machine

S: máquina

See *system.*

man

F: homme

S: hombre

See *user.*

man-machine communication

F: communication homme-machine

S: comunicación hombre-máquina

The interchange of data between *user* and *system.*

man-machine interface

F: interface homme-machine

S: interfaz hombre-máquina

The set of *inputs, outputs,* and special *actions* as well as the man-machine interaction mechanism, including *diaglogue procedures* and the interrelationships identified for these entities in the various *functional areas.*

man-machine language (MML)

F: langage homme-machine (LHM)

S: lenguaje hombre-máquina (LHM)

The means of expression used in communication between the *user* and the *system.*

man-machine terminal

F: terminal homme-machine

S: terminal hombre-máquina

An *input/output device,* that enables the *user* and the *system* to communicate with each other, e.g., visual display terminal, printer.

manual response

F: réponse manuelle

S: respuesta manual

A *user* response to a *system* invitation that may comprise the actuation of keys on *terminals* or switch frames, replacement of equipment, etc.

menu

F: menu

S: menú

A list of items, from which a selection can be made by the *user.*

menu identity

F: identité de menu

S: identidad de menú

An identity unique to a *menu* so that it can be distinguished from other *menus.*

menu item

F: rubrique de menu

S: elemento de menú

A brief description of an item in a *menu,* optionally accompanied by a *selection identity,* in order to allow a choice to be made by inputting such an identity.

menu-item selection

F: sélection en mode menu

S: selección de elemento de menú

The activity of selecting an item using the item selection procedure and the repetition of this activity for subsequent *menus* until ultimately the procedure results in something other than further *menu output.*

menu output

F: sortie de menu

S: salida de menú

An *output* of a *menu,* used in *information entry* procedures.

message waiting indication

F: indication de message en instance

S: indicación de mensaje en espera

A means of announcing, within a *dialogue procedure,* the presence of a high priority *output* addressed to this *man-machine terminal.*

meta-language

F: métalangage

S: metalenguaje

Formal means of representation using defined *symbols* according to specific rules.

methodology (for the specification of the man-machine interface)

F: méthodologie (pour la spécification de l'interface homme-machine)

S: metodología (para la especificación del interfaz hombre-máquina)

A five-phase general working procedure that (1) provides for the generation of *MML function semantics* and (2) provides for the creation of an actual *man-machine interface* using *syntax, dialogue procedures,* and *MML function semantics.*

MML

F: LHM

S: LHM

See *man-machine language.*

MML function

F: fonction LHM

S: función LHM

See *class A function.*

MML function decomposition

F: subdivisión de fonction LHM

S: descomposición de función LHM

The division of a *function* into its constituent parts.

MML function semantics

F: sémantique de fonction LHM

S: semántica de función LHM

Semantics peculiar to one or more *MML functions* within the *functional areas* (or sub-areas) that were generated by the application of the *methodoly* for the specification of the *man-machine interface.* It is based upon *actions, objets, information entities* and their interrelationships.

MML syntax and dialogue procedures meta-language

F: syntaxe et métalangage de procédure de dialogue LHM

S: metalenguaje de sintaxis y de procedimiento de diálogo del LHM

A graphical *meta-language* for representing MML *input* and *output syntax* as well as *dialogue procedures.*

monologue output

F: sortie de monologue

S: salida de monólogo

Output from the *system* which occurs outside a *dialogue.*

multi-line (subscriber line)

F: multiligne (ligne d'abonné)

S: línea de abonado multilínea

A line between a public *exchange* and a P(A)BX or a line between a public *exchange* and a subscriber set belonging to a *subscriber line group.*

Named-defined parameter

F: paramètre défini par nom

S: parámetro definido por el nombre

A *parameter* which is identified by its *parameter name.*

network

F: réseau

S: red

All the *exchanges* which are relevant from the service standpoint operated by an Administration in a country.

network element

F: élement de réseau

S: elemento de red

Telecommunication equipment which may perform signalling, switching and transmission functions.

network group

F: groupe de réseaux

S: grupo de redes

A group of telecommunication networks relevant to the service standpoint (e.g. different operating companies offering the same service in the same country).

network management action

F: action de gestion du réseau

S: acción de gestión de red

The activity performed, not necessarily in the *network elements*, to regulate traffic flow.

network management centre

F: centre de gestion du réseau

S: centro de gestión de red

A centre where network management functions are performed (e.g., 0 and M centre, switching centre).

network management control

F: commande de gestion du réseau

S: control de gestión de red

The capabilities in *network elements* to regulate traffic flow and network operation in order to insure the maximum utilization of the network capacity in all situations of traffic overload and *network element* failure.

network management data

F: données de gestion du réseau

S: datos de gestión de red

The set of information necessary to monitor, detect and identify a network problem.

network management indicator

F: indicateur de gestion du réseau

S: indicador de gestión de red

Logical result of the comparison of *Network Management Parameters* and thresholds comparison.

network management information

F: information de gestion du réseau

S: información de gestión de red

The set of information produced in the *network management centre* describing the network status and performance, the abnormal conditions detected, the problems identity and the active *network management controls.*

network management object

F: objet de gestion du réseau

S: objeto de gestión de red

A set of *network elements* under control of network management *functions* and/or subject to measurement for network purposes.

network management parameters

F: paramètres de gestion du réseau

S: parámetros de gestión de red

Information produced in the *network management centre* to be used for the production of *Abnormal Condition Report* and for display on alerting devices.

network management system

F: système de gestion du réseau

S: sistema de gestión de red

A system which performs network management functions.

network problem identity

F: identité de problème du réseau

S: identidad de problema de red

Information produced in the *network management centre* to indicate the type of problem detected and the portion of the network and/or services affected.

network raw data

F: données brutes de réseau

S: datos de red en bruto (sin procesar)

Network information provided by *network elements* and used for the production of *Network Management Parameters* and for display on alerting devices.

network reference data

F: données de référence du réseau

S: datos de referencia de la red

Information on the *network elements* and structure (e.g., *circuit groups*, number of *circuits* in a circuit group, routing information, type and quantity of switching system *components).*

non-decimal numeral

F: numéral non décimal

S: numeral no decimal

A *numeral* in a *numbering system* other than *decimal.*

non-terminal symbol

F: symbole non terminal

S: símbolo no terminal

Representation, within a *syntax diagram*, of another *syntax diagram* by name. It is an abbreviated *symbol* for a more complex construct.

numbering system

F: système de numération

S: sistema de numeración

Any notation for the representation of numbers.

numeral

F: numéral

S: numeral

A discrete representation of a number within a *numbering system.*

objet

F: objet

S: objeto

An *information entity*, usually the *system* part towards which the *action* of a *function* is directed.

octal numeral

F: numéral octal

S: numeral octal

A *numeral* in the octal (base 8) *numbering system*, represented by the characters 0 (zero), 2, 3, 4, 5, 6, 7, optionally preceded by O' (letter O apostrophe).

on-line documentation

F: documentation en ligne

S: documentación en línea

A comprenhensive body of information provided a user on-line about a given subject related to a *function.*

on-line help

F: assistance en ligne

S: ayuda en línea

See *solicited guidance.*

on-line training

F: formation en ligne

S: adiestramiento en línea

A comprenhensive body of information provided a user on-line to supplement or replace other training methods such as classroom instruction, training manuals or video courses.

operation and maintenance system

F: système d'exploitation et de maintenance

S: sistema de operación y mantenimiento

A *system* which supports administration personnel in performing operation and maintenance *jobs* related to *SPC systems.*

operational procedure

F: procédure d'exploitation

S: procedimiento operacional

A process illustrating the interrelationship of *user* and *system* in performing an operation, maintenance, installation or acceptance testing *job.*

Operation and Maintenance Centre (OMC)

F: centre d'exploitation et de maintenance (CEM)

S: centro de operación y mantenimiento (COM)

A physical location staffed by administration personnel responsible for operation and maintenance (O&M) of *SPC systems.*

other information

F: autre information

S: otra información

General information which may accompany the *function models* and the lists of *MML functions* in the documents B and C.

output

F: sortie

S: salida; extraer

i) Information that is transferred from the *system* to the *user*, e.g., *help output*, etc.

ii) An *action* to transfer specified data from the *system* to a *man-machine terminal.*

output field

F: champ de sortie

S: campo de salida

See *inaccessible field.*

output outside dialogue

F: sortie hors dialogue

S: salida fuera de diálogo

A spontaneous *output* indicating a certain event, e.g., an alarm situation, or an output in response to a *command* previously entered in an *interactive operating sequence*, e.g., a traffic measurement result.

output parameters

F: paramètres de sortie

S: parámetros de salida

Data determining *output* routing and scheduling.

output and input window area

F: sous-fenêtre de sortie et d'entrée

S: zona de ventana de salida y de entrada

These two *window areas* should support scrolling and should be user controllable in size. The input *window area* should be used for *direct information* entry. Response to the *direct information entry* as well as *output outside dialogue* should appear in the output *window area.* Input acknowledgements may also appear directly following the command in the input *window area.* The *scrolling* should occur in *two window areas* separately, or both *window areas* may be combined into one *window area.*

parameter

F: paramètre

S: parámetro

Data which identifies and contains pieces of information necessary to execute a *command.*

parameter argument

F: argument de paramètre

S: argumento de parámetro

The smallest portion of a *parameter value* which specifies an appropriate object or value. It can be a *simple* or a *compound* structure and may be used singularly or as part of a group.

parameter block

F: bloc de paramètres

S: bloque de parámetros

See *block of parameters.*

parameter block entry sequence

F: séquence d'introduction d'un bloc de paramètres

S: secuencia de introducción de bloque de parámetros

A procedure used to input a *block of parameters.*

parameter block request indication

F: indication de demande de bloc de paramètres

S: indicación de petición de bloque de parámetros

An indication from the *system* to the *user* to proceed with *input* of *parameters.*

parameter identity

F: identité de paramètre

S: identidad de parámetro

The *parameter label* and optional *parameter position* identifying a *parameter* in a form.

parameter label

F: étiquette de paramètre

S: etiqueta de parámetro

A *text string* used in *forms* to identify a *parameter.*

parameter name

F: nom de paramètre

S: nombre de parámetro

An *identifier* which indicates unambiguously the meaning and structure of the subsequent *parameter value.*

parameter position

F: position de paramètre

S: posición de parámetro

The sequence number of a *parameter* in a *block of parameters* or in a *form.*

parameter value

F: valeur de paramètre

S: valor de parámetro

The part of a *parameter* that contains the information required to specify any appropriate object(s) or value(s). It consists of one or a group of *parameter arguments.*

parameter value input field

F: champ d'entrée de valeur de paramètre

S: campo de entrada de valor de parámetro

An *accessible field* that is normally empty or filled in by the *system* and should be filled in or overwritten by the *user.*

password

F: mot de passe

S: contraseña

A character string used for identification and authorization of a *user.*

periodicity pattern

F: schéma de periodicité

S: esquema de periodicidad

A pattern which indicates which days are recording (or results *output*) days and which are not. The start day positions this time span. Once activated, the execution of the measurements (or of the results *output*) is performed according to this pattern, until disabled by a deactivation *command.*

phase

F: phase

S: fase

One of five steps of the general working produce that forms the *methodology* for the specification of a *man-machine interface.*

position-defined parameter

F: paramètre défini par position

S: parámetro definido por la posición

A *parameter* whose nature is identified by its position in the *parameter block* of a*command.*

procedure body

F: corps de procédure

S: cuerpo de procedimiento

That part of a *dialogue procedure* where *commands* can be entered and new physical areas can be addressed, dependent on the authority of the *user.*

procedure description

F: description de procédure

S: descripción de procedimiento

A method of representing an *operational procedure.*

procedure epilogue

F: épilogue de procédure

S: epílogo de procedimiento

The procedure used to terminate the *dialogue procedure.* It consists of an *action* by the *user* to deactivate the *dialogue* and/or an *output* from the *system* to indicate the *end of dialogue.*

procedure prologue

F: prologue de procédure

S: prólogo de procedimiento

A set of *actions* to activate the *man-machine terminal,* to call the *system* and to identify the *user.*

prompting

F: proposition

S: sugerencia

A method used by the *system* to request *input* from the *user* in a *dialogue procedure.*

promption output

F: sortie de proposition

S: salida de sugerencia

An *output* from the *system* providing guidance on the next *input* requirement.

ready indication

F: indication «prêt»

S: indicación de preparado

An *output* element used in a *dialogue procedure* to indicate that the direction of the dialogue has changed and that the *system* is ready to receive a *command* or a *destination identifier.* It is also used as the *identification invitation.*

ready indicator

F: indicateur «prêt»

S: indicador de preparado

An *indicator* used in the *ready indication* to indicate that the *system* is ready to receive information.

recording

F: enregistrement

S: registro

Performance of the operations implied by the measurement entities in order to collect the required data.

recording day

F: jour d'enregistrement

S: día de registro

Day when a *recording* is performed. Several *recording periods* are allowed within a *recording day.* No overlap of *recording periods* is allowed for the same measurement. Each *recording period* can have a different length.

recording period

F: période d'enregistrement

S: periodo de registro

A period of *recording* during a *recording day.*

rejection output

F: sortie de rejet

S: salida de rechazo

An *output* message indicating that an *input* to the *system* is valid and will not be acted upon, and corrections cannot be applied.

relational operator

F: opérateur de relation

S: operador relacional

An operator (see Table 2/Z.314) in a *selection argument.* Used in data base queries.

remove

F: retirer

S: retirar

An *action* to request the *system* to take specified equipment units out of service; the system still retains knowledge of the units so that they may be returned to service by the *restore* action.

request

F: demande

S: petición

A manual *action* used to activate a *man-machine terminal* and the *system.*

request output

F: sortie de demande

S: salida de petición

A type of *response output* requesting further *input action* from the *user,* e.g., correction of an erroneous *parameter,* or supplying further information.

response output

F: sortie de réponse

S: salida de respuesta

An *output* message in the *dialogue procedure* which gives information about the state of an *input.* The output can beany one of the following types: *acceptance output, rejection output* and *request output.*

restore

F: rétablir

S: restablecer

An *action* to return specified equipment units to service; opposite of *remove.*

results accumulutation period

F: période d'accumulation des résultats

S: período de acumulación de resultados

Time interval within a *recording period* during which the required measurement entities are processed and at the end of which results are stored for immediate or later output.

results output routing

F: acheminement de la sortie des résultats

S: encaminamiento de salida de resultados

Data defining the media to which results *output* is to be directed.

results output schedule

F: calendrier de sortie de résultats

S: calendario de salida de resultados

Data specifying a set of days (or a periodicity pattern) and of times during these days when the output of the results is to be made.

route

F: acheminement

S: encaminar

An *action* to instruct the system that any subsequent *output* of a certain type should be routed to specified media.

route

F: route

S: ruta

Collection of *circuit sub-groups* between two *exchanges* that are equivalent for routing purposes. The term route in Recommendation Z.335 is equivalent to the concept of "*Circuit Group*" as used in Recommendation Z.337 and in E-Series Recommendations.

route group

F: groupe de voies (d'acheminement)

S: grupo de rutas

The set of all the possible *routes* on which a call may be forwarded to the appropriate destination.

scrolling

F: défilement

S: desplazamiento vertical

The ability to display the part of the data not currently visible in the *window area.*

selection

F: sélection

S: selección

A *symbol* of the *decomposition meta-language* which indicates that the choice among several *information entities* is possible.

selection argument

F: argument de sélection

S: argumento de selección

An argument comprising one or more conditions. Used in data base queries.

selection identity

F: identité de sélection

S: identidad de selección

An identity unique to a *menu item* so that it can be distinguished from other *menu items* within the same *menu.*

semantics

F: sémantique

S: semántica

The rules and conventions governing the interpretation and assignment of meaning to constructions in a language.

separator

F: séparateur

S: separador

A character used to delimit *syntax* elements.

sequence

F: séquence

S: secuencia

A *symbol* of the *decomposition meta-language* which indicates a left-to-right ordering of *information entities.*

session

F: session

S: sesión

See *dialogue procedure.*

session status

F: état de session

S: estado de sesión

Information reflecting the current status of the *session* in terms of *user* identity, destination identity, etc.

set

F: positionnement

S: poner

An *action* to place equipment units in a specified state (number of possible states greater than 2); possible states include in service and out of service.

single line (subscriber line)

F: ligne individuelle (ligne d'abonné)

S: línea de abonado unilínea; línea individual

A line between a public *exchange* and a subscriber set.

simple parameter argument

F: argument de paramètre simple

S: argumento de parámetro simple

A *parameter argument* made up of only one *information unit.*

sollicited guidance

F: guidage sollicité

S: orientación solicitada

System's capability to provide a user with information on how to use the system while using it.

sort

F: trier

S: clasificar

An *action* to rearrange the order of a *data set* according to specified (or default) criteria; the contents of the original set is not affected by this *action*, only its order.

source identifier

F: identificateur d'origine

S: identificador de origen

One or more *information units* indicating the physical area where an *output* was generated.

SPC system

F: système SPC

S: sistema CPA

See *stored program controlled (SPC) system.*

special keys and directives information window area

F: sous-fenêtre d'information sur les touches spéciales et les directives

S: zona de ventana de información sobre teclas especiales y directrices

This *window area* should display *function key* labels and specifies about the use of *directives.*

Specification and Description Language (SDL)

F: langage de description et de spécification (LDS)

S: lenguaje de especificación y descripción

The specification and description language specified in the Z.100-series Recommendations.

spontaneous menu

F: menu spontané

S: menú espontáneo

A *menu* that is automatically given at the start of an *information entry.*

spontaneous output

F: sortie spontanée

S: salida espontánea

An *output* generated by internal events of the *system,* e.g., an alarm.

start date

F: date de début

S: fecha de comienzo

Start day for the measurement execution.

start time

F: heure de début

S: hora de comienzo

Time for beginning the *recording period* in a *recording day.*

status window area

F: sous-fenêtre d'état

S: zona de ventana de estado

This *window area* should contain alarm indicators of the system being controlled, trouble reporting information from connected equipment, and message waiting indicators.

stop date

F: date de fin

S: fecha de terminación

Stop day for the mesurement execution.

stop time

F: heure de fin

S: hora de terminación

Time for terminating a *recording period* in a *recording day.*

stored program controlled (SPC) system

F: système de commande par programme enregistré (SPC)

S: sistema de control por programa almacenado (CPA)

A system (this includes switching *systems*) that provides telecommunication services.

subdivision

F: subdivision

S: subdivisión

A symbolic means in the *decomposition meta-language* of indicating the division of an entity into its constituent parts.

subscriber line group

F: groupe de lignes d'abonné

S: grupo de líneas de abonado

A group of line groups which are recognized and managed by a public *exchange* as a logical group.

supplementary information

F: information supplémentaire

S: información adicional

Information presenting an explanation to the *user* if required so as to ease the *input* of the *parameter value.*

symbol

F: symbole

S: símbolo

A conventional representation of a concept or a representation of a concept upon which agreement has been reached.

symbolic name

F: nom symbolique

S: nombre simbólico

A character string used for the representation of an entity.

syntax

F: syntaxe

S: sintaxis

The rules for the formation of permissible constructions (e.g., character strings) in a language, without regard to meaning.

syntax diagram

F: diagramme de syntaxe

S: diagrama sintáctico

A representation either of the syntactic structure of the construct or of a portion of the *dialogue procedure.*

system

F: système

S: sistema

Computer-based equipment and the *applications* used in telecommunications to provide service to the subscriber or to support administration personnel in their *jobs.*

system information

F: information du système

S: información del sistema

Information related to the status of the system. It may contain items such as: system status *indicators,* alarm *indicators,* and a *message waiting indicator.*

table

F: tableau

S: cuadro; talla

An ordered presentation of interrelated information.

terminal

F: terminal

S: terminal

Abbreviation for *man-machine terminal.*

terminal symbol

F: symbole terminal

S: símbolo terminal

A *symbol* containing a character or string of characters which actually appear in the *input* or *output.*

terminology harmonization

F: harmonisation de la terminologie

S: armonización de la terminología

Standardization of the terminlogy to be used in the generation of *MML function semantics.*

text block

F: bloc de texte

S: bloque de texto

Any combination of *clarifying texts, named-defined parameters* and/or *tables* which gives *output* information wherever it is needed or requested.

text string

F: chaîne de texte

S: cadena de texto

A character string (excluding " (quotation mark) and *correction characters*) not interpreted within the *man-machine language* but stored in the *system* for later *output* in its original form.

tool

F: outil

S: instrumento

A means by which the task of one or more *phases* of the *methodology* for the specification of the *man-machine interface* may be accomplished.

user

F: usager

S: usuario

The human being in *man-machine communication.*

user guidance

F: guidage de l'usager

S: orientación del usuario

Información displayed by the *system* to help the *user* to perform the task.

variable text

F: texte variable

S: texto variable

A string of *information units* which contains information unique to the event caused the *output.*

video attributes

F: attributs vidéo

S: atributos video

Attributes to distinguish certain important information (e.g., a title, a message, a chosen item) in order to attract the attention of the *user.* They work on the characters of the information shown within an entire *window,* a part of a *window area,* an entire *field* or within a part of a *field.*

visible display

F: zone visible

S: zona visible

The entire physical screen of a visual display terminal.

window

F: fenêtre

S: ventana

A *window* is a collection of one or more *window areas.* Collection depends on the *application.* A *window* is dedicated to an application.

window area

F: sous-fenêtre

S: zona de ventana

A *window area* is a named part of a *window* (sometimes the entire window) that is dedicated for a specific purpose depending upon the *application.*

work window area

F: sous-fenêtre de travail

S: zona de ventana de trabajo

This *window area* should be used for *information entry through form filling* and *information entry through menu-item selection.* The *window area* may also be used as a graphic display and screen editor area, and should support scrolling.

ANNEX A

(to Recommendation Z.341)

Classification of terms

A.1 *Introduction*

This annex classifies the MMI terms according to the following classification scheme.

The purpose of this classification is to partition MMI terms into conceivable collections:

a) to users of the MMI Recommendations;

b) to those who can benefit from the knowledge of MMI terms, such as programmers; and

c) to developers of the MMI Recommendations, for the identification of new work items, the organization of future work and new Recommendations.

The classes are intended to aid the identification and delimitation of the scope of the terms and therefore enhance the readability of the Recommendations and specifications. For example, an MMI designer who wants to document the terms to be used at the MMI can use this classification to organize the MMI documentation. The following defines MMI terms and describes their use.

This classification scheme is inspired by the conceptual scheme work [1] within ISO. The application area and scope are, however, different.

A.2 *Overview*

The totality of MMI terms is divided into classes shown in Figure A-1/Z.341. The classes can form a hierarchical structure. If a term is applicable to more than one class, it may appear in the most general class.

The classes can very well be divided into subclasses.

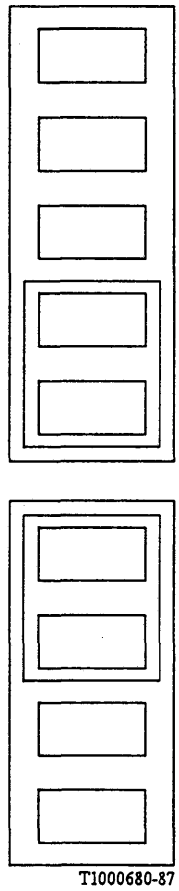


FIGURE A-1/Z.341

Classification scheme

A.3 *Classes and their usage*

The following text defines different classes, explains their usage and gives examples of their contents. Terms are classified as application- independent and application-dependent terms. These classes are classified into subclasses.

1) *Application-independent terms*

Terms which can be common for all MMI applications:

a) Primitives

Terms which cannot be further defined but are used to define other terms.

Users: Everyone, in particular, language designers. End users can read system documentation which applies these terms.

Examples: Predicates, logical connectors, quantifiers, terms, etc.

b) Metaterms

General terms which are not restricted to the classes listed below.

This class includes common terms from mathematics and logic as well as general terms such as time and space.

Users: Everyone, in particular, language designers. End users can read system documentation which applies these terms.

Examples: Arithmetics, algebra, etc.
Space, time, terminology, phenomenon, etc.

c) System specification terms

Terms restricted to one system description language, often aimed at one application area.

Users: Everyone, in particular, applications system designers including MMI designers. End users can read system documentation which applies these terms.

Example: Block channel, signal, etc. n basic SDL.
Flow line, non-terminal input symbol, etc., in the MML metalanguage.
Entity, relationship, domain, cardinality, etc., in the Entity-relationship approach.

d) Application-independent MMI manipulation and presentation terms.

Terms needed to manipulate all MMI terms and to present terms common for all MMI applications.

d1) MMI manipulation terms

Terms which can be used to manipulate other terms.

Users: Everyone who applies a man-machine interface, both end users and system developers.

Examples: Next page, delete window, etc.
Retrieve (from database to screen), insert (from screen to database).

d2) MMI presentation terms

Terms which are needed to present other terms.

Presentation terms are intended to present other terms.

Users: Everyone who observes a man-machine interface, both end users and system developers.

Examples: Window, window area, field, pixel, etc.
Common field, work window area, etc.

2) *Application-dependent terms*

Terms which are not common for all MMI applications.

e) MMI terms for operations and maintenance

e1) computer equipment and computer software

Terms used for the management of softward and equipment through their whole life cycle.

Users: Everyone who explicitly accesses, installs or supports computer resources, or administers their availability including access administration.

Examples: Machine, terminal, program, program statement database, etc.
Logon, backup, suspend, etc.

e2) telecommunications networks and services'

Terms for the Administrations' manipulation and presentation of telecommunications networks and services.

Users: Everyone in the Administrations who administers telecommunications networks or services.

Examples: Subscriber, multiplex group, traffic intensity, installation plan, subscriber equipment, etc.
Subscriber identity, circuit identity, etc.
Insertion of routing data, etc.

f) MMI terms for the use of telecommunications networks and services

Terms which are specific for the subscribers' manipulation and presentation of telecommunications networks and services.

Users: Everyone who applies telecommunications networks or services.

Examples: Dial up, send Teletex, electronic envelope, etc.

g) Other application-dependent terms

Any application-dependent term which is not listed above.

Users: Everyone who applies MMIs for manipulating or presenting data concerned with the actual topics.

Examples: Employee number, salary information, etc.

A.4 *Classification of terms*

1) *Application-independent terms*

b) application metaterms

application

arithmetic delimiter

arithmetic operator

arithmetic expression

binary numeral

decimal numeral

digit

graphic characters

hexadecimal numeral

keyed numeral

letter

non-decimal numeral

numbering system

numeral

octal numeral

semantics

symbol

syntax

c) System specification terms

annotation

annotation symbol

Bachus Naur Form (BNF)

character set

component

composit part

connectivity rules

connector
decomposition meta-language
default option
documents A through G
drawing convention
flowline
general option
guidelines
information entity
information structure (diagram)
information structure meta-language
iteration
meta-language
methodology (for the specification of the man-machine interface)
MML function decomposition
MML function semantics
MML syntax and dialogue procedures meta-language
non-terminal symbol
object
other information
phase
procedure description
selection
separator
sequence
Specification and Description Language (SDL)
subdivision
syntax diagram
terminal symbol
terminology harmonization

d) Application-independent MMI manipulation and presentation terms
CCITT MML
man-machine communication
man-machine interface
man-machine language (MML)
MML

d1) MMI manipulation terms
action
action modifier
activate
browse
change
command
command code
command entry sequence
continuation character
control character

control functions
control key
correction character
create
cursor control functions
deactivate
delete
dialogue
dialogue element
dialogue procedure
direct information entry
directive
edit
error correction
escape indication
execution character
filter
form filling
format effector
function
function key
indicator
information entry
information entry through form filling
information entry through menu-item selection
inhibit
initialize
input
interactive
interactive operating sequence
interface control functions
interrogate
item selection procedure
manual response
menu-item selection
MML function
operational procedure
output
parameter block entry sequence
procedure body
procedure epilogue
procedure prologue
scrolling
session
sort

d2) MMI presentation terms

acceptance input
acceptance output
accessible field
additional header information
additional information
block of parameters
border area
clarifying text
command reference
command sequence number
comment
compound parameter argument
concealment
cursor
data set
default value
delimiter
display area
end of dialogue
end of output
end statement
field
form
form identity
form output
general information window area
guidance output
header
help output
highlighting
identification invitation
identifier
inaccessible field
information unit
input acknowledgement
input error
input error information
input field
input window area
interaction request output
inverse video
item description
layout option
menu
menu identity
menu item

menu output
message waiting indication
monologue output
name-defined parameter
on-line documentation
on-line help
on-line training
output and input window area
output field
output outside dialogue
parameter
parameter argument
parameter block
parameter block request indication
parameter identity
parameter name
parameter position
parameter value
parameter value input field
password
position-defined parameter
prompting
prompting output
ready indication
ready indicator
rejection output
request output
response output
selection identity
session status
simple parameter argument
solicited guidance
special keys and directives information window area
spontaneous menu
spontaneous output
status window area
supplementary information
symbolic name
table
text block
text string
user guidance
variable text
video attributes
visible display
window
window area
work window area

2) *Application-Dependent Terms*

e) MMI terms for operations and maintenance

e1) Computer equipment and computer software

block mode transmission
character mode transmission
destination identifier
destination prologue
graphic terminals
I/O device
man-machine terminal
remove
request
restore
route
route
set
source identifier
terminal
tool

e2) Telecommunication networks

administrative system
alarm statement
allow
auxiliary system
class A function
class B function
class C function
exchange
functional area (or sub-area)
function model
job
job area
machine
operation and maintenance system
Operation and Maintenance Centre (OMC)
SPC system
Stored Program Control (SPC) system
system
system information

f) MMI terms for the use of telecommunications networks and services

g) Other application-dependent terms

man
user

References

[1] J.J. VAN GRIETHUYSEN, ed., *Concepts and Terminology for the Conceptual Schema and the Information Base*, Report ISO/TC97/SC21-N197, ANSI, 1982.

ISBN 92-61-03811-5