CHAPTER 3

SPECIFICATION OF THE HUMAN-COMPUTER INTERFACE

LANGUAGE

The human-computer 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 various activities of telecommunications systems. Consideration of these functions has been an essential prerequisite for the development of the CCITT based man-machine language (MML).

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 semantic relevant to the human-computer interface.

3.1 BASIS OF MML

Man-machine language (MML) is the means of exchanging
information between users and systems. The MML contains features which are sufficient to ensure that all relevant functions for the operation, maintenance, administration can be performed.

The basic attributes of the language are as follows:

1. 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.

2. 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.

3. The MML is adaptable to different kinds of personnel and to different national language and organisational requirements.

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

The MML is sufficiently flexible to meet Administration's requirements for the organization of their operation and maintenance staff and for the security of their systems; it does 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 actions will not cause the system to stop, unduly alter the system configuration or take up undue resources.

3.2 INPUT (COMMAND) LANGUAGE SYNTAX SPECIFICATION

3.2.1 Command Structure

3.2.1.1 Command -

A command begins with the command code, which define the function to be performed by the system. If further information is required a command code can be followed by a parameter part consists of one or more blocks of parameters. A command is always completed by an execution character.

3.2.1.2 Command Code -

The command code is composed of up to three indentifiers 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.

3.2.1.3 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.

3.2.1.4 Parameters -

A parameter identifies and contains a piece of information and may be either position defined or parameter name defined.

3.2.1.5 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.

3.2.1.6 Name Defined Parameter -

A 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.

3.2.1.7 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.

3.2.1.8 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 is applied a parameter value reduces to a parameter argument.

3.2.1.9 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. A parameter argument consists of a simple or a compound parameter argument.

3.2.1.9.1 Simple Parameter Argument -

A simple parameter argument consists of one information unit.

3.2.1.9.2 Compound Parameter Argument -

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

3.2.1.10 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. However the default base for a keyed numeral cannot be overwritten by another base.

3.2.1.11 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.
3.2.1.11.1 Grouping Of Blocks Of Parameters -

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

3.2.1.11.2 Grouping Of Parameter Arguments -

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

3.2.1.11.2.1 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). 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&&9++2&10 means the simple parameter arguments 5, 7, 9 and 10.

3.2.1.11.2.2 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). 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.

An explicit increment value can be specified following the
oper 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, 8-3.

1.3 OUTPUT STRUCTURE

1.3.1 Output Outside Dialogue

The output described is output outside dialogue. This output is either a spontaneous output indicating a certain event or it is a delayed response to an interactive operating sequence.

1.3.2 Header

The header is given in output outside dialogue. It is also used in the dialogue procedure. 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 the input output function can be added to the header as additional header information.
3.3.3 Format Effector

A format effector is used to format output in a suitable manner. Certain format effectors are specifically incorporated in the output definition.

3.4 HUMAN-COMPUTER DIALOGUE PROCEDURES

Human-computer communication comprises two types of information interchange, namely dialogue and output outside dialogue; they occur sequentially and in no particular order.

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 section. In the text the terms "dialogue" and "dialogue procedure" are used interchangeably.

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.

3.4.1 Definition Of The Dialogue Procedure
3.4.1.1 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. The dialogue can be terminated by a procedure epilogue.

3.4.1.2 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 eligibility of the user to issue command;
- a header, which is given from the system and contains the workstation identification, information relating to date and time, etc.

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 an interactive operating sequence.

The request, the identification of the user and the header are defined in the following paragraphs.
3.4.1.2.1 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.

3.4.1.2.2 Identification Procedure -

The identification procedure is used for identification and authorization of a user. The identification can provide access to groups of commands which can have different security or functional classifications. The identification invitation may request the user to identify himself by means of a password. The password must be input following a ready indication.

3.4.1.2.2.1 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).

3.4.2 Header Display

The header is output by the system at the end of the procedure prologue.
3.4.2.1 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 consists of keying a sequence of characters on the keyboard.

3.4.2.2 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 sequence or special actions. The latter occurs when, as a result of partial execution of a function, the system requests the supply of further information the form of special actions or further commands for which human judgement and/or decision is required.

3.4.2.3 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. ABORT, etc.
3.4.2.4 Interaction Request Output -

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

3.4.3 End Statement

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

3.4.3.1 Direct Parameter Input -

Only one method of inputting parameters is dealt with in direct parameter input.

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 response output.

If terminated by an execution character and responded by acceptance or rejection output, the system concludes the direct parameter input. If terminated by a continuation character and responded by acceptance or rejection output, the system is required to return a parameter block request indication that functions as in indication to proceed with the input of the next block or blocks of parameters. If responded by request output the system is required to return
a parameter block request indication that functions as in 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 abort command function.

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

3.4.3.2 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 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:

1. Guidance output followed by a help. 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.

2. 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 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.

3.4.3.3 Parameter Block Request Indication -

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

3.4.4 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.

3.4.4.1 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.

3.4.4.2 Rejection Output -

Rejection output is an indication by the system that the input received is not valid and will 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.

3.4.4.3 Request Output -

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

Man-machine language syntax has been described in Appendix B.