5i’

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 | ists the operation, maintenance, installation and acceptance testing functions to be controlled by means of the MML.

Recommendation Z.332 | resents 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 | resents 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*
| 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;

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

— 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.

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*

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

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.

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, p.

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.

Figure I-2/Z.331 [T1.331], p. (Traiter comme tableau MEP)

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 to 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 ´different types of functions is shown in Figure 1/Z.332.

Figure 1/Z.332, p.

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.

Document C

- Other information (whenever applicable).

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.

Document B

— Other information (whenever applicable).

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.

Document C

— Other information (whenever applicable).

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.

Document F

— An evaluation of the preliminary

man-machine interface.

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, p.4

Figure 3/Z.332, p.5

Figure 4/Z.332, p.6

Recommendation Z.333

METHODOLOGY FOR THE SPECIFICATION OF THE | fR 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

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.

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

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);

— 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 | | ;

— 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, p.

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, p.

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:

Figure, p.

3.4.1.2.2 *Component*

A component is an information entity that is not structured further.

The following symbol is used:

Figure, p.

3.4.1.3 *Structuring*

3.4.1.3.1 *Subdivision*

Subdivision in Information Structure Diagrams is shown in the following way:

Figure, p.

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:

Figure, p.

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:

Figure, p.

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 $n < m$.

Figure, p.

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:

Figure, p.

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:

Figure, p.

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:

Figure, p.

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* .

Figure, p.

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.

Figure, p.

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.

Figure, p.

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.

Figure, p.

For the selection symbol, in case of a high number of possibilities, the following drawing conventions are also allowed:

Figure, p.

Where the number of information entities in a structure is undetermined, this could be shown as

Figure, p.

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, p.24

Figure II-1b/Z.333, p.25

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

numera } { &<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 }

MONTAGE: REC. Z.334 A LA FIN DE CETTE PAGE