INTERNATIONAL TELECOMMUNICATION UNION

ITU-T Q.1213

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(10/95)

INTELLIGENT NETWORK

GLOBAL FUNCTIONAL PLANE FOR INTELLIGENT NETWORK CS-1

ITU-T Recommendation Q.1213

(Previously “CCITT Recommendation”)

FOREWORD

The ITU-T (Telecommunication Standardization Sector) is a permanent organ of the International Telecommunication Union (ITU). The ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 1 (Helsinki, March 1-12, 1993).

ITU-T Recommendation Q.1213 was prepared by ITU-T Study Group 11 (1993-1996) and was approved under the WTSC Resolution No. 1 procedure on the 17th of October 1995.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

CONTENTS

SUMMARY

IN capability set 1 (IN CS-1) is the first standardized stage of the intelligent network as an architectural concept for the creation and provision of telecommunication services. This Recommendation provides the Intelligent Network (IN) Global Functional Plane (GFP) architecture for IN capability set 1 (IN CS-1). This Recommendation defines the IN GFP for IN CS-1 based on the general framework for IN GFP provided in Recommendation Q.1203, consistent with the scope of IN CS-1 defined in Recommendation Q.1211.

This Recommendation defines:

- the IN GFP model for IN CS-1;
- identifications and definition of the 15 IN CS-1 service independent building-blocks (SIBs), including the Basic Call Process (BCP) specialized SIB;
- the use of global service logic in IN CS-1.

Companion Recommendations include the Q.1200 and Q.1210 Series Recommendations.

Recommendation Q.1213

GLOBAL FUNCTIONAL PLANE FOR INTELLIGENT NETWORK CS-1

(Helsinki, 1993; modified in 1995)

1 General

The concepts for the Intelligent Network (IN) are embodied in the Intelligent Network Conceptual Model (INCM) as described in associated Recommendations I.312/Q.1201 [1]. The Global Functional Plane (GFP) of the INCM is described in associated Recommendations I.329/Q.1203 [2].

2 Scope of IN global functional plane for capability set 1

IN Capability Set 1 (IN CS-1) is the first standardized stage of the intelligent network as an architectural concept for the creation and provision of telecommunication services. This Recommendation provides the functional characteristics of the GFP associated specifically with IN CS-1. General GFP aspects are addressed in associated Recommendations I.312/Q.1201 [1] and I.329/Q.1203 [2].

The following functional characteristics are specific to IN CS-1 and are addressed in this Recommendation:

- IN CS-1 GFP Modelling (see clause 4).
- IN CS-1 SIBs (see clause 5).

  For IN CS-1, 14 Service Independent Building Blocks (SIBs) are specified (not including the BCP SIB discussed below). These represent the minimum set of SIBs required to define the IN CS-1 targeted services, identified in Recommendation Q.1211 [3].
- BCP SIB (see clause 6).

  For IN CS-1 the basic call process has been defined as a specialized SIB which provides the basic call capabilities. Nine Points of Initiation (POIs) for Global Service Logic (GSL) interfaces have been specified. In addition, six Points of Return (PORs) for GSL interfaces have been specified.
- Plane-to-plane mapping (see clause 8).

  The relationships between the service plane and the GFP are specified for IN CS-1.
- For IN CS-1, network time is assumed to be available to all SIBs in the GFP and it does not have to be passed through call instance data.
- Stage 1 SDL diagrams are provided for IN CS-1 SIBs only where the SDLs clarify the understanding of the SIB operation.

3 References

The following Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time, the editions indicated were valid. All Recommendations and other references are subject to revision: all users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published.

[1] CCITT Recommendation I.312/Q.1201 (1992), *Principles of intelligent network architecture.*

[2] CCITT Recommendation I.329/Q.1203 (1992), *Intelligent network global functional plane architecture*.

[3] ITU-T Recommendation Q.1211 (1993), *Introduction to intelligent network capability set 1*.

[4] ITU-T Recommendation Q.1290 (1993), *Glossary of terms used in the definition of intelligent networks*.

[5] CCITT Recommendation I.130 (1988), *Method for the characterization of telecommunication services supported by an ISDN and network capabilities of an ISDN*.

[6] ITU-T Recommendation Z.100 (1993), *CCITT specification and description language (SDL)*.

[7] ITU-T Recommendation Q.931 (1993), *Digital subscriber Signalling System No. 1 (DSS 1) – ISDN user-network interface layer 3 specification for basic call control*.

4 IN CS-1 global functional plane modelling

4.1 Role of SIBs in the global functional plane

SIBs are abstract representations of network capabilities that exist in an IN structured network. Their definition enforces the concept of service and technology independence by decoupling the services from the technology on which services are provided. The 14 SIBs defined in this Recommendation identify the network capabilities which will be available to service designers in a CS-1 IN structured network. Their use will provide valuable insight on how service creation will be facilitated in CS-n Recommendations.

4.2 Additional characteristics of a IN CS-1 SIB

Data required by each SIB is defined by SIB support data parameters and call instance data parameters.

SIBs are global in nature and their location need not be considered as the whole network is regarded as a single entity in the GFP.

4.3 IN CS-1 global functional plane model

The INCM described in Recommendation I.312/Q.1201 [1] models the GFP as shown in Figure 1. Normal or non-IN supported services are processed within the BCP. When an IN supported service is to be invoked, its GSL is launched at the POI by triggering mechanism from the BCP. The global functional plane models the network from a global, or a network wide, point of view. Contained in this view is the BCP SIB, including IN CS-1 POIs and PORs, the IN CS-1 SIBs, and the GSL, which describes how SIBs are chained together to describe service features.

This Recommendation specifies how the model can be used to fully define IN CS-1 service features in the GFP.

4.4 Terminology

The following terminology used in this Recommendation, is more fully defined in Recommendation Q.1290 [4]:

CID Call Instance Data

CIDFP Call Instance Data Field Pointer

CLI Calling Line Identification

SSD Service Support Data

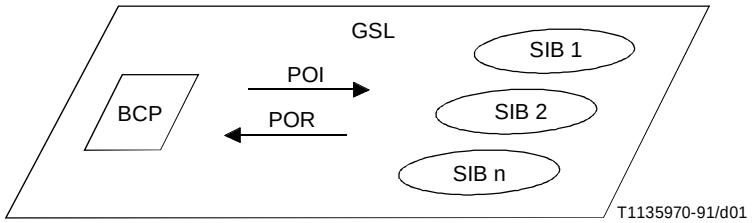

FIGURE 1/Q.1213

Global functional plane model

5 IN CS-1 Service Independent Building Blocks (SIBs)

The following list of SIBs have been identified as required to support the list of targeted IN CS-1 services and service features identified in Recommendation Q.1211 [3]:

- ALGORITHM
- AUTHENTICATE
- CHARGE
- COMPARE
- DISTRIBUTION
- LIMIT
- LOG CALL INFORMATION
- QUEUE
- SCREEN
- SERVICE DATA MANAGEMENT
- STATUS NOTIFICATION
- TRANSLATE
- USER INTERACTION
- VERIFY

The stage 1 SIB descriptions that follow reflect the understanding of the logical function of each SIB in its role of supporting IN CS-1 services and service features. The BCP, which is viewed as a specialized SIB, is described in clause 6.

An explanation of the format of the stage 1 SIB descriptions is outlined below.

5.1 Data parameters for SIBs

By definition, SIBs are independent of the service/SF they are used to represent. They have no knowledge about previous or subsequent SIBs which are used to describe the service feature.

In order to describe service features with these generic SIBs, some elements of service dependence is needed. Service dependence can be described using data parameters which enable a SIB to be tailored to perform the desired functionality. Data parameters are specified independently for each SIB and are made available to the SIB through global service logic.

Two types of data parameters are required for each SIB, dynamic parameters called Call Instance Data (CID) and static parameters called Service Support Data (SSD).

5.1.1 Call Instance Data (CID)

Call instance data defines dynamic parameters whose value will change with each call instance. They are used to specify subscriber specific details like calling or called line information. This data can be:

- made available from the BCP SIB (e.g. Calling Line Identification);
- generated by a SIB (e.g. a translated number); or
- entered by the subscriber (e.g. a dialled number or a PIN code).

Associated with each CID value is a logical name which is referred to as the CID Field Pointer (CIDFP). If a SIB requires CID to perform its function, there will be an associated CIDFP assigned through SSD (refer to 5.1.2). For instance, the TRANSLATE SIB's CID which defines what is to be translated is called Information. The TRANSLATE SIB's SSD parameter which defines where this data can be found is CIDFP-Filter Value(s).

Since the CID value can vary with each call instance, service features can be written with data flexibility. In the above TRANSLATE SIB example, one service feature may require translation of a calling number, while another service feature will require translation of the called number. In both cases, the data required by the SIB is specified by the information Calling Line Identity (CLI), but the CIDFP-Filter Value(s) changes. In the first service feature, the value of CIDFP-Filter Value(s) is set to CLI, while the second service feature sets the value of CIDFP-Filter Value(s) to Called Number.

Once a CIDFP has been specified for a service feature, it can be referenced by subsequent SIBs, and the CID value can be made available to all subsequent SIBs in the SIB chain. This CIDFP is said to be fixed for that service and is constant for all instances of that service. The actual value of the CID changes for each call instance of that service feature.

5.1.2 Service Support Data (SSD)

Service support data defines data parameters required by a SIB which are specific to the service feature description. When a SIB is included in the GSL of a service description, the GSL will specify the SSD values for the SIB. SSD consists of:

i) Fixed parameters

These are data parameters whose values are fixed for all call instances. For instance, the "File Indicator" SSD for the TRANSLATE SIB needs to be specified uniquely for each occurrence of that SIB in a given service feature. The "File Indicator" SSD value is then said to be fixed, as its value is determined by the service/SF description, not by the call instance.

If a service/SF is described using multiple occurrences of the same SIB, then fixed SSD parameters are defined uniquely for each occurrence.

ii) Field pointers

Field pointers identify which CID is required by the SIB, and in doing so provide a logical location for that data. They are signified by "CIDFP-xxxx", where "xxxx" names the data required. For instance, "CIDFP-Filter Value(s)" for the TRANSLATE SIB will specify which CID element is to be translated.

If more than one CID is required by a SIB to perform its function, then the SSD data parameters will contain multiple field pointers.

5.2 Method to describe IN CS-1 SIBs

The following template applies to characterize and describe the SIBs.

5.2.1 Definition

Prose description of the SIB from the service creation point of view.

5.2.2 Operation

Description of actions performed by the SIB. The operations section expands on the definition, to allow the reader to clearly understand the operation that this SIB is intended to perform.

5.2.3 Potential service applications

Service examples of where this SIB can be used.

5.2.4 Input

Input to each SIB is specified as three distinct elements:

- one logical start;
- Service Support Data which defines parameters which are specified by the service description;
- Call Instance Data which are specific to that call instance.

5.2.5 Output

Output from each SIB is specified as two distinct elements:

- one or more logical end;
- call instance data which defines data parameters specific to that call instance which results from the execution of that SIB and are required by other SIBs or the BCP to complete the call service instance.

5.2.6 Graphic representation

A graphic representation is used to describe input, operations and output of the SIB and is illustrated in Figure 2. Each SIB is characterized by having one logical start and one or more logical ends. These logic flows are shown by the solid arrows on the left and right of the diagram. Each logic flow is specified above each arrow. SSD parameters are identified by the dashed arrow at the top of the diagram and are specified beside the dashed arrow. Similarly, CID parameters are specified below the diagram. Input CID parameters are separated from the output parameters.

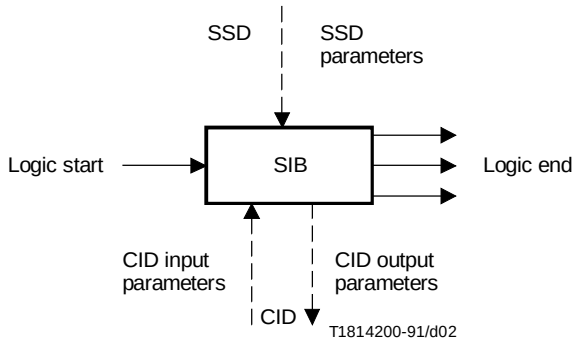

FIGURE 2/Q.1213

Graphical representation of IN CS-1 SIB

5.2.7 SDL diagram

This diagram gives a graphic representation of the stage 1 description of the SIB using SDL macro diagrams (see Recommendation Z.100 [6]).

5.3 ALGORITHM

5.3.1 Definition

Applies a mathematical algorithm to data to produce a data result.

5.3.2 Operation

This SIB takes a specified call instance data and applies the specified mathematical algorithm to it to produce the corresponding data result.

It can be used to implement a simple arithmetic operation as incrementing a counter.

5.3.3 Potential service applications

– Mass calling;

– Televoting.

5.3.4 Input

5.3.4.1 Logical start

Indicates the logical start of execution for the SIB.

5.3.4.2 Service support data

– Type

Specifies the type of algorithm for this SIB. Two algorithms have been identified for IN CS-1:

a) increment;

b) decrement.

– Value

Specifies the amount to be used when applying the SIB (e.g. 1, 2, etc.).

– CIDFP-Data

This CID field pointer specifies the call instance data to which the algorithm is to be applied.

– CIDFP-Error

This CID field pointer specifies where in output call instance data the error cause will be written.

5.3.4.3 Call instance data

– Data

The identifier is the data associated with the CIDFP-Data upon which algorithm is to be applied.

5.3.5 Output

5.3.5.1 Logical end

– Success;

– Error.

5.3.5.2 Call instance data

– Data

Contains the resultant value after the SIB has completed.

– Error cause

Identifies the specific condition which caused an error during the operation of the SIB. The following errors have been identified for ALGORITHM:

a) invalid type;

b) invalid value.

5.3.6 Graphical representation

See Figure 3.

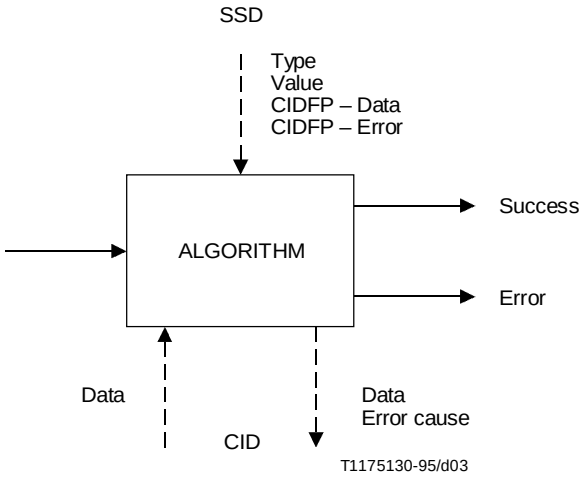

FIGURE 3/Q.1213

ALGORITHM SIB

5.4 AUTHENTICATE

5.4.1 Definition

Provides authentication functionality for a service.

5.4.2 Operation

This SIB provides an authentication function to establish an authorized relationship between the service logic and a database on behalf of a user.

The authentication function can be either simple authentication or externally defined.

5.4.3 Potential service applications

– All services.

5.4.4 Input

5.4.4.1 Logical start

Indicates the logical start of execution for the SIB.

5.4.4.2 Service support data

– Authenticate Name (optional)

This SSD field is only used if SSD Authenticate Mechanism ID has value “Simple”. The Authenticate Name is used to identify the location of the object containing the Authenticate Password attribute in order to set up an Authorized Relationship. A “null” in this SSD field indicates that the Authenticate Name can vary and its value must be provided through call instance data.

– Authenticate Password (optional)

This SSD field is only used if SSD Authenticate Mechanism ID has value “Simple”. The Authenticate Password provides the data to be matched in the Authenticate Name object in order to set up an Authorized Relationship. A “null” in this SSD field indicates that the Authenticate Name can vary and its value must be provided through call instance data.

– Authenticate Mechanism Id

The Authenticate Mechanism Id identifies the authentication mechanism which is to be used to establish the Authorized Relationship. It may take one of the following values:

- None, which implies that no authentication is performed.
- Simple, which implies that simple authentication is performed.
- External, which implies that an externally defined authentication is performed.

– CIDFP – Error

This CID field pointer specifies where in output call instance data the error cause will be written.

– CIDFP – Authorized Relationship Id

This CID field pointer specifies where in output call instance data the Authorized Relationship Id will be written.

– CIDFP – Authenticate Name

This CID Field Pointer specifies which call instance data is to be used as the Authenticate Name.

– CIDFP – Authenticate Password

This CID Field Pointer specifies which call instance data is to be used as the Authenticate Password.

5.4.4.3 Call instance data

– Authenticate Name (optional)

This CID field is only used if SSD Authenticate Mechanism Id has value “Simple” and SSD Authenticate Name has value “null”. The Authenticate Name is used to identify the location of the object containing the Authenticate Password attribute in order to set up an Authorized Relationship.

– Authenticate Password (optional)

This CID field is only used if SSD Authenticate Mechanism ID has value “Simple” and SSD Authenticate Name has value “null”. The Authenticate Password provides the data to be matched in the Authenticate Name object in order to set up an Authorized Relationship.

NOTE – When Authenticate Name and Authenticate Password are supplied as input CID, the SSD parameters Authenticate Name and Authenticate Password are not used.

5.4.5 Output

5.4.5.1 Logical end

– Success (Authentication successful);

– Error.

5.4.5.2 Call instance data

– Error cause

Identifies the specific condition which caused an error during the operation of the SIB.

– Authorized Relationship Id

The Authorized Relationship Id provides the identity of the established Authorized Relationship through which operations can be applied.

5.4.6 Graphical representation

See Figure 4.

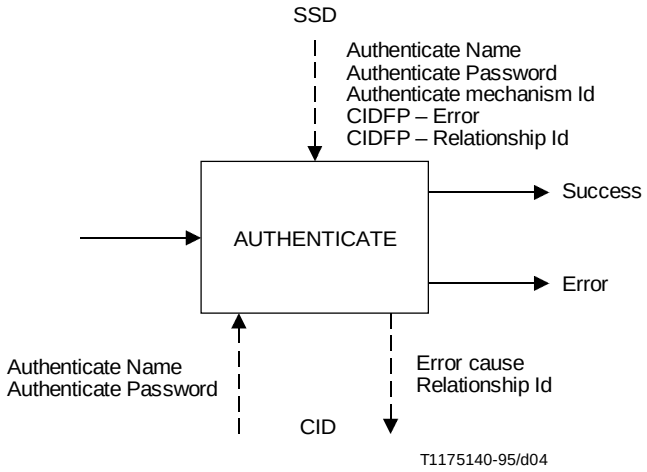

FIGURE 4/Q.1213

AUTHENTICATE SIB

5.5 CHARGE

5.5.1 Definition

Determine special charging treatment for the call, where special refers to any charging in addition to that normally performed by the basic call process.

In general, this involves identifying:

- the resources for which charging is to occur;
- to where the charges are to be directed.

It should be noted that this SIB is not responsible for the subscriber billing process.

5.5.2 Operation

The charging information output by this SIB must be compatible with the charging and billing system of the network operator or service provider. However, this SIB does not define the format of the output nor does it identify all the types of information that an Administration will require for charging. Generation of bills will often be done off-line by the Administration's existing billing system.

The CHARGE SIB is used for specific resource charging and may be invoked several times in one service/service feature instance.

Different simultaneous or consecutive call instances may charge the same account.

Typical resources for which charging can occur are:

- circuit-mode bearers;
- packet or messages;
- SRF resources, e.g. announcements, voice message storage, etc.;
- SCF usage (in units).

Typically, charging may be directed towards:

a) the account identified by the CLI;

b) the account identified by the called number (either the dialled number or the destination number);

c) an account or credit card identified by the collected user information;

d) the calling user's exchange accumulator;

e) a payphone.

5.5.3 Potential service application

– Any service which requires specific IN charging.

5.5.4 Input

5.5.4.1 Logical start

Indicates the logical start of execution for the SIB.

5.5.4.2 Service support data

– number of accounts to charge;

– Account.

Each account is specified by two parameters, as follows:

a) Number

Specifies one of the following:

1) CIDFP-Line

This CID field pointer specifies which call instance data is to be used as the line number to charge.

2) CIDFP-Account

This CID field pointer specifies which call instance data is to be used as the account number to charge.

3) Fixed account

Specifies an account number which is fixed for all call instances.

b) Per cent (%)

Specifies the allocation of the total charge for this account. The sum of all allocations must equal 100%.

– Resource type

Specifies the resource to be charged for (e.g. bearer type, announcement, SCF usage, etc.).

– Units

Specifies a premium value for the specified resource type.

– Service/service feature identifier

Specifies the service/service feature for which charging is to be applied.

– CIDFP-Pulse

This CID field pointer specifies which call instance data is used to identify pulse metering on the calling line.

– CIDFP-Error

This CID field pointer specifies where in output call instance data the error cause will be written.

5.5.4.3 Call instance data

– Line(s)

Specifies a line number for charging. This can be the calling line, the dialled number or a destination number.

– Account(s)

Specifies an account number for charging. This is an account number which was entered during the call such as a credit card or a calling card number.

– Pulse metering

When specified, indicates that pulse metering is associated with the calling line.

5.5.5 Output

5.5.5.1 Logical end

1) Success;

2) Error.

5.5.5.2 Call instance data

– Error cause

Identifies the specific condition which caused an error during the operation of the SIB. The following errors have been identified for CHARGE:

a) invalid account to charge;

b) invalid percentage allocation (> 100, < 0);

c) invalid sum of percentage allocations (not equal to 100);

d) invalid service/service feature;

e) invalid resource;

f) invalid units;

g) system fault - unable to write record.

5.5.6 Graphical representation

See Figure 5.

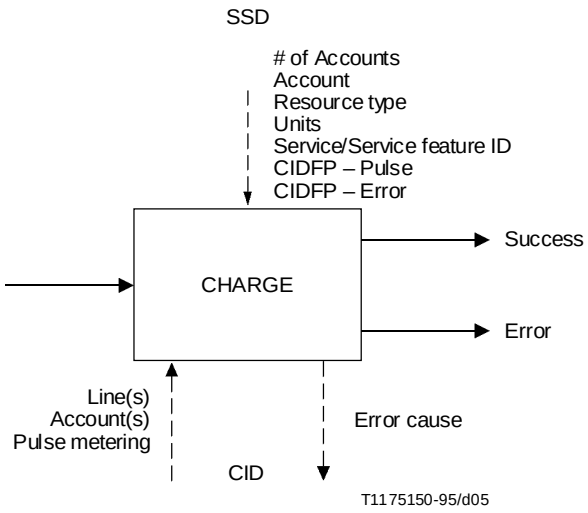

FIGURE 5/Q.1213

CHARGE SIB

5.6 COMPARE

5.6.1 Definition

Performs a comparison of an identifier against a specified reference value. Three results are possible:

– identifier is GREATER THAN the value;

– identifier is LESS THAN the value;

– identifier is EQUAL TO the value.

5.6.2 Operation

This SIB compares an identifier to a specified reference value. One of three logical ends will result from this operation (e.g. <, >, or =). Other logical ends can be formulated by combining two of the logical ends together (e.g. < >, <=, or =>).

It can be used for:

a) Comparing an identifier to a specified reference value. For instance, for checking that the current number of calls is less than the maximum number authorized.

b) For checking the relationship of current network time to a customer specified time to perform time dependent decision. The comparison may be done on Time of Day (TOD), Day of Week (DOW), or Day of Year (DOY). The reference value is then the customer specified TOD, DOW or DOY.

5.6.3 Potential service applications

– Time dependent routing;

– CCBS.

5.6.4 Input

5.6.4.1 Logical start

Indicates the logical start of execution for the SIB.

5.6.4.2 Service support data

– Comparison type

Specifies the type of comparison to be performed. The following types have been identified:

a) Identifier value – Compare the identifier against the reference value.

b) Time – Compare network time to the reference value. Network time is specified as:

1) time of day;

2) day of week; or

3) day of year.

– CIDFP-Data

This CID field pointer specifies which call instance data is to be used as the identifier.

– Reference value

Specifies the value against which the comparison will be made.

– CIDFP-Error

This CID field pointer specifies where in output call instance data the error cause will be written.

5.6.4.3 Call instance data

– Data

Specifies the data associated with the CIDFP-Data (see SSD) which is to be compared to the reference value.

5.6.5 Output

5.6.5.1 Logical end

– GREATER THAN the value;

– LESS THAN the value;

– EQUAL TO the value;

– Error.

5.6.5.2 Call instance data

– Error cause

Identifies the specific condition which caused an error during the operation of the SIB. The following errors have been identified for COMPARE:

a) invalid identifier;

b) invalid reference value.

5.6.6 Graphical representation

See Figure 6

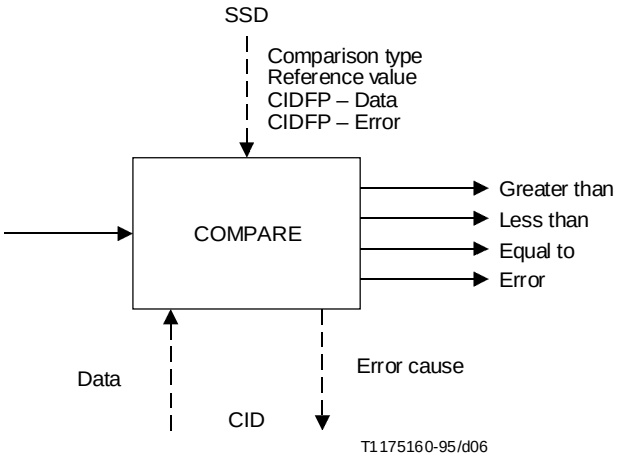

FIGURE 6/Q.1213

COMPARE SIB

5.7 DISTRIBUTION

5.7.1 Definition

Distribute calls to different logical ends of the SIB based on user specified parameters.

5.7.2 Operation

This SIB distributes calls to its different logical ends based on a user identified algorithm. For example, calls could be distributed based on a percentage allocated to each logical end.

5.7.3 Potential service applications

– mass calling;

– televoting;

– freephone.

5.7.4 Input

5.7.4.1 Logical start

Indicates the logical start of execution for the SIB.

5.7.4.2 Service support data

– Algorithm type

a) percentage;

b) sequential;

c) time of day;

d) day of week.

NOTE 1– Hierarchical call distribution may be realized through the use of multiple instances of the DISTRIBUTION SIB in conjunction with the STATUS NOTIFICATION SIB.

– Number of logical ends

NOTE 2 – Unlike most SSD, change of this value will effect the structure of the GSL this SIB is contained in.

– Algorithm parameters

If Type = percentage

- For each logical end
  - percentage (sum over all paths must equal 100).

If Type = sequential

- None

If Type = time of day

- For each logical end
  - begin time associated with this logical end;
  - end time associated with this logical end.

  All 24 hours of the day must be accounted for, with no overlap of time across logical ends.

If Type = day of week

- For each logical end
  - all days of the week to be associated with this logical end.

  All days of the week must be accounted for. No day may be associated with more than one logical end.

– CIDFP-Error

This CID field pointer specifies where in output call instance data the error cause will be written.

5.7.4.3 Call instance data

– None

5.7.5 Output

5.7.5.1 Logical end

– 1

– 2

– ...

– N [where N = number of logical ends (item 1 of input SSD)]

– Error

5.7.5.2 Call instance data

– Error cause

Identifies the specific condition which caused an error during the operation of the SIB. The following errors have been identified for DISTRIBUTION:

a) invalid type;

b) invalid percentage allocation (> 100, < 0);

c) invalid sum of percentage allocations (not equal to 100);

d) invalid number of logical ends;

e) missing time period;

f) overlap of the time across logical ends;

g) missing days;

h) overlap of days across logical ends.

5.7.6 Graphical representation

See Figure 7.

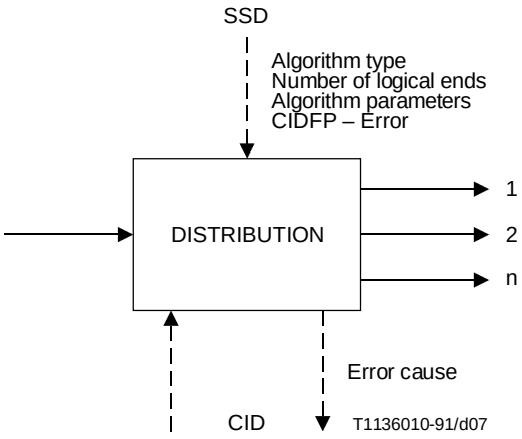

FIGURE 7/Q.1213

DISTRIBUTION SIB

5.8 LIMIT

5.8.1 Definition

Limit the number of calls related to IN provided service features. Such limiting will be based on user specified parameters.

NOTE – This SIB is not used for network congestion management functions.

5.8.2 Operation

This SIB may be used to pass all or a fraction of all calls related to IN provided service features.

For example, calls may be passed:

- for a specific duration (which may be infinite) at specific intervals;
- according to a counting algorithm (e.g. pass N calls out of P).

The LIMIT SIB in conjunction with other SIBs (e.g. COMPARE) can provide the required functionality for time dependent call limiting (e.g. TOD).

5.8.3 Potential service applications

- mass calling;
- televoting;
- freephone.

5.8.4 Input

5.8.4.1 Logical start

Indicates the logical start of execution for the SIB.

5.8.4.2 Service support data

- Type

  Specifies the type of limiting to be performed. The following types have been identified:

  a) pass calls for (S) seconds out of every (Q) seconds;

  b) pass (N) calls out of (P) calls.

- Parameters

  Specifies the parameters associated with the SSD type selected above:

  1) S duration;

  2) Q interval;

  3) P counter interval;

  4) N number of calls.

- CIDFP-File

  This CID field pointer specifies where the current limit count is located.

- CIDFP-Error

  This CID field pointer specifies where in output call instance data the error cause will be written.

5.8.4.3 Call instance data

- File

  Identifies the data associated with CIDFP-File which specifies the current count of calls.

5.8.5 Output

5.8.5.1 Logical end

- pass;
- no pass;
- error.

5.8.5.2 Call instance data

– Error cause

Identifies the specific condition which caused an error during the operation of the SIB. The following errors have been identified for LIMIT:

a) invalid type;

b) invalid parameter.

5.8.6 Graphical representation

See Figure 8.

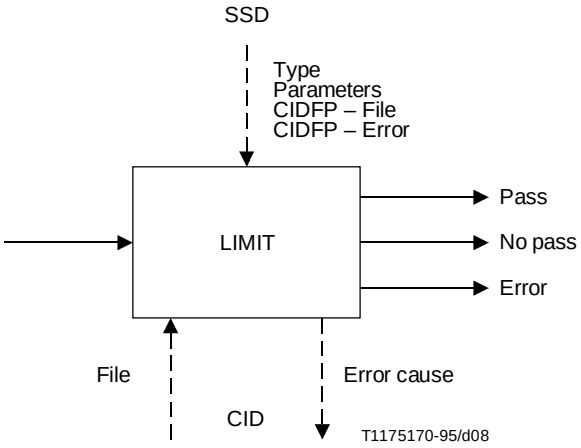

FIGURE 8/Q.1213

LIMIT SIB

5.9 LOG CALL INFORMATION

5.9.1 Definition

Log detailed information for each call into a file. The collected information may be used by management services (e.g. statistics, etc.) and not by call-related services.

5.9.2 Operation

This SIB logs (or writes) call related information to a specified file. The type of call-related information to be logged will be identified by service support data. Each instance of this SIB will cause the recording of the specified CID information.

5.9.3 Potential service applications

– All services.

5.9.4 Input

5.9.4.1 Logical start

Indicates the logical start of execution for the SIB.

5.9.4.2 Service support data

– CIDFP-Data

These CID field pointers specify which call instance data are to be used as identifiers. CIDFP-Data can include:

a) call attempt time;

b) call stop time;

c) call connect time;

d) dialled number;

e) destination number;

f) additional dialled number (e.g. credit card number, etc.);

g) calling line identification;

h) time in queue;

i) bearer capability;

j) error causes;

k) any other CID.

– Log file name

Specifies the log file object where the value of the identifier is to be logged.

– CIDFP - Authorized Relationship Id

This CID field pointer specifies which call instance data is to be used as the Authorized Relationship Id.

– CIDFP-Error

This CID field pointer specifies where in output call instance data the error cause will be written.

5.9.4.3 Call instance data

– Data

Specifies the data associated with the CIDFP-Data (see SSD) which is to be logged.

– Authorized Relationship Id

The Authorized Relationship Id provides the identity of the established Authorized Relationship through which the operations can be applied.

5.9.5 Output

5.9.5.1 Logical end

– Success (information written);

– Error.

5.9.5.2 Call instance data

– Error cause

Identifies the specific condition which caused an error during the operation of the SIB. The following errors have been identified for LOG CALL INFORMATION:

- invalid identifier;
- invalid log file indicator.

5.9.6 Graphical representation

See Figure 9.

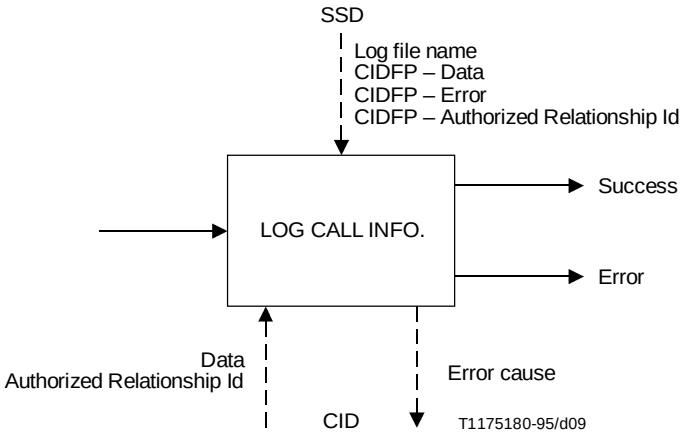

FIGURE 9/Q.1213

LOG CALL INFO SIB

5.10 QUEUE

5.10.1 Definition

Provide sequencing of IN calls to be completed to a called party.

5.10.2 Operation

This SIB provides all the processing needed to provide queueing for a call, and will specifically:

- pass the call if resources are available;
- queue the call;
- optionally play announcements to caller on queue;
- when resources become available, dequeue the call.

5.10.3 Potential service application

- All IN services which use call queueing service feature.

5.10.4 Input

5.10.4.1 Logical start

Indicates the logical start of execution for the SIB.

5.10.4.2 Service support data

- Max active

  Specifies the maximum number of active calls allowed for the resource.

- Max number

  Specifies the maximum number of calls allowed on queue at a given time.

- Max time

  Specifies the maximum time the call may remain on the queue.

– Announcement parameters

Specify the control values for announcements. The control values which can be specified are:

- Announcement ID: specifies which announcement is to be sent; the Announcement ID could be "null" to signify that no announcement is to be sent.
- Repetition requested: specifies if the announcement is to be repeated;
- Repetition interval: specifies the delay period in seconds between repetitions;
- Maximum repetitions: specifies the maximum number of times the announcement will be repeated.

– CIDFP-Call

This CID field pointer specifies which call instance data identifies the specific call.

– CIDFP-Resource

This CID field pointer specifies which call instance data identifies the resource.

– CIDFP-Error

This CID field pointer specifies where in output call instance data the error cause will be written.

5.10.4.3 Call instance data

– Call reference

Identifies the specific call which is a candidate for queueing.

– Resource

Specifies the data associated with the CIDFP-Resource which identifies the resource for which the call will be queued.

5.10.5 Output

5.10.5.1 Logical end

1) Resource available;
2) Call party abandon;
3) Queue timer expiry;
4) Queue full;
5) Error.

5.10.5.2 Call instance data

– Time spent in queue

Identifies the total time that a particular call was queued.

– Error cause

Identifies the specific condition which caused an error during the operation of the SIB. The following errors have been identified for QUEUE:

a) invalid Max active;
b) invalid Max number;
c) invalid Max time;
d) invalid announcement parameters;
e) invalid call reference.

5.10.6 Graphical representation

See Figures 10 and 11.

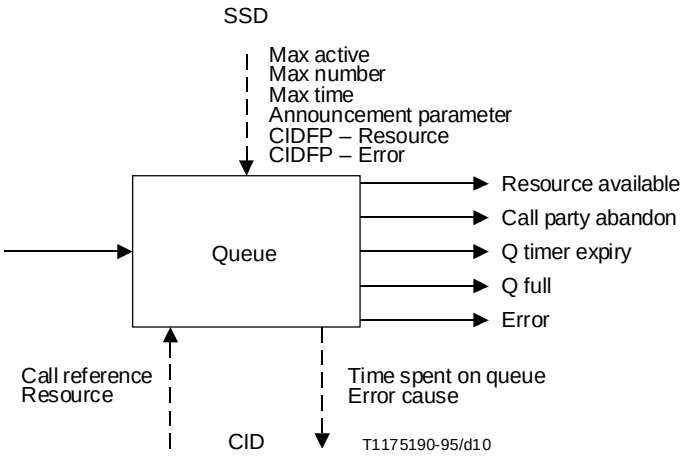

FIGURE 10/Q.1213

QUEUE SIB

5.11 SCREEN

5.11.1 Definition

Perform a comparison of an identifier against a list to determine whether the identifier has been found in the list.

5.11.2 Operation

This SIB takes the appropriate identifier and determines whether or not it is contained within the list identified by the SIB support data. A "Match" condition results if the identifier has been found on the list.

Examples using the SCREEN SIB are:

– for verifying a user ID or PIN;

– for terminating or originating call screening based on a network address.

Multiple instances of the SCREEN SIB, in conjunction with other SIBs, like TRANSLATE and COMPARE, may result in more complex screening capabilities.

5.11.3 Potential service applications

– Selective call forward on busy/don't answer;

– Originating call screening;

– Terminating call screening;

– Security screening;

– Account card calling;

– Credit card calling.

5.11.4 Input

5.11.4.1 Logical start

Indicates the logical start of execution for the SIB.

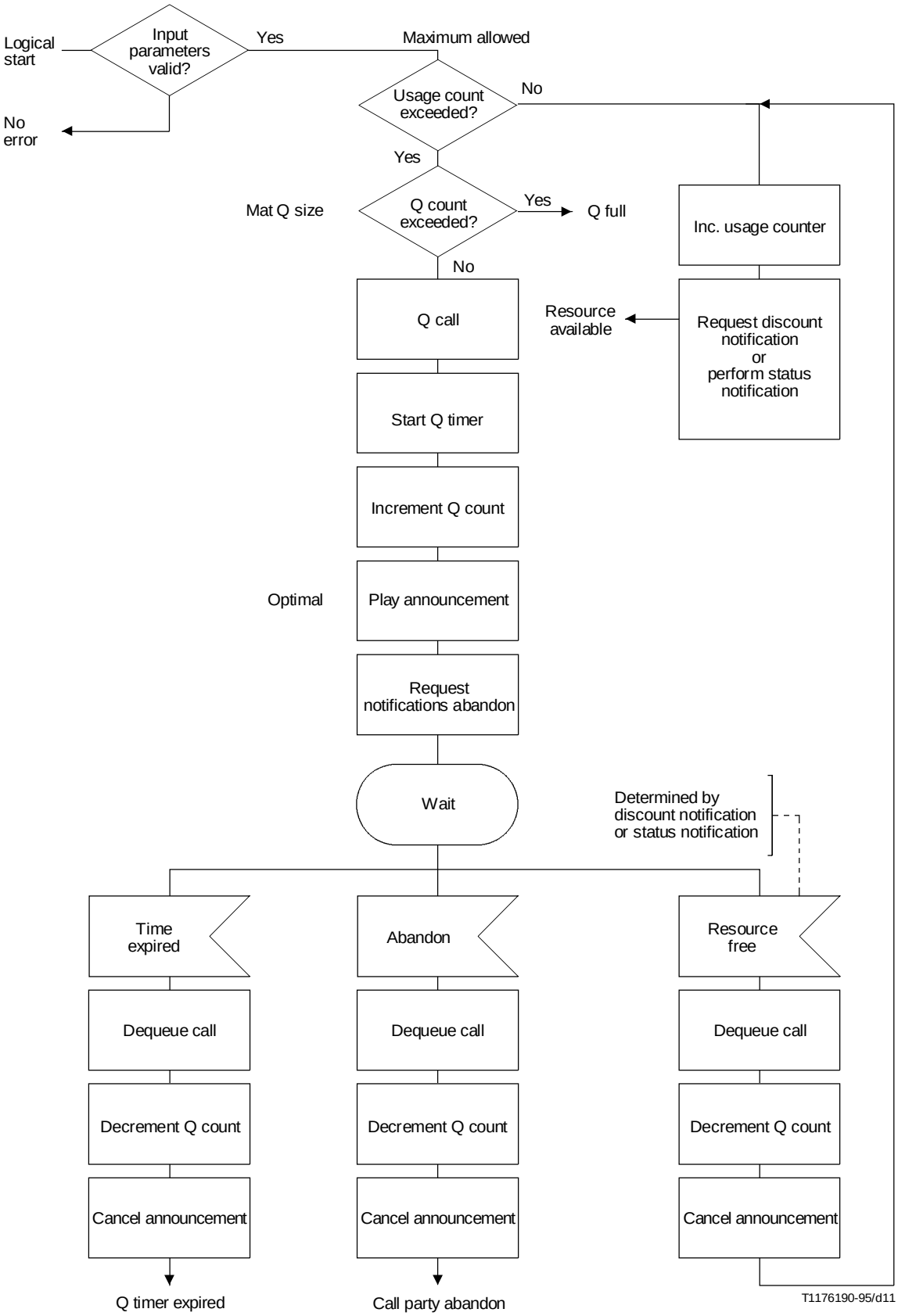

FIGURE 11/Q.1213

QUEUE SIB stage 1

5.11.4.2 Service support data

– Screen list name

The screen list name identifies the screen data object to be used.

– Screen list Filter

The Screen list Filter identifies the attributes and filter tests to be applied to the attributes in the screen data object. The attribute values are stored in the CIDFP – Screen pointers.

– CIDFP-Data

This CID field pointer(s) specifies which call instance data is to be used as the identifier(s).

– CIDFP - Authorized Relationship Id

This CID field pointer specifies which call instance data is to be used as the Authorized Relationship Id.

– CIDFP-Error

This CID field pointer specifies where in output call instance data the error cause will be written.

5.11.4.3 Call instance data

– Data

The Data CID is the attributes values associated with the CIDFP-Data (see SSD) which is to be filtered against the screen data object.

– Authorized Relationship Id

The Authorized Relationship Id provides the identity of the established Authorized Relationship through which the operations can be applied.

5.11.5 Output

5.11.5.1 Logical end

– Match [= on the list].

– CIDFP – Authorized Relationship Id

This CID field pointer specifies which call instance data is to be used as the Authorized Relationship Id.

– CIDFP-Error

This CID field pointer specifies where in output call instance data the error cause will be written.

5.11.5.2 Call instance data

– Error cause

Identifies the specific condition which caused an error during the operation of the SIB. The following errors have been identified for SCREEN:

a) invalid identifier;

b) invalid screen list.

5.11.6 Graphical representation

See Figure 12.

5.12 SERVICE DATA MANAGEMENT

5.12.1 Definition

Enables end user specific data to be replaced, retrieved, incremented, or decremented.

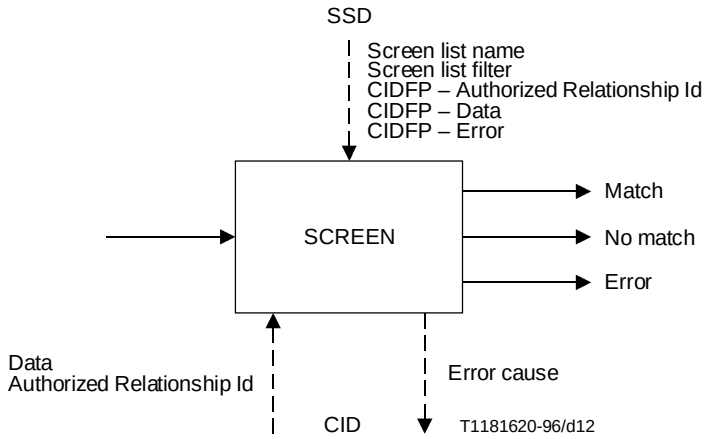

FIGURE 12/Q.1213

SCREEN SIB

5.12.2 Operation

This SIB performs the appropriate actions, i.e. replace, retrieve, increment, or decrement information stored within the network. For example, this SIB could be used to retrieve or replace a customer's call forwarding number.

5.12.3 Potential service application

– Call forwarding;

– Customer profile management.

5.12.4 Input

5.12.4.1 Logical start

Indicates the logical start of execution for the SIB.

5.12.4.2 Service support data

– Object name

Specifies the subscriber data object to be used.

– Action

Specifies the operation to be performed on the subscriber data. The following actions are allowed:

a) Replace – Replace the existing data element in the subscriber file specified by the element indicator with new data specified by the information value.

b) Retrieve – Copy the data element specified by the element indicator and place it in the output CID called data retrieved.

c) Increment – Increase the value of the data element in the subscriber file specified by the element indicator by the amount indicated by the Inc/Dec value.

d) Decrement – Decrease the value of the data element in the subscriber file specified by the element indicator by the amount indicated by the Inc/Dec value.

e) SetToDefault – Reset the value of the data attribute in the subscriber data object to its default value.

f) addObject – Add a new data object into the database. In this case the Attribute indicator must contain all of the object attributes which are required to correctly initialize the data object.

g) removeObject – Remove a data object from the database.

– Attribute indicator

Specifies the data attribute in the subscriber data object upon which the action is to be performed. This optional data parameter is only required when the field to be acted upon is constant for all call instances (e.g. changing a call forwarding number). A "null" in this SSD field indicates that the attribute indicator can vary and its value must be provided through call instance data.

– Inc/Dec value

Specifies the amount by which the attribute indicator is to be incremented or decremented.

– CIDFP-Info

This CID field pointer specifies which call instance data is to be used as the information value.

– CIDFP-Attribute

This CID field pointer specifies which call instance data is to be used as the attribute indicator. If the attribute indicator is to be specified by the SSD, then this CID field pointer will not be used by the SIB.

– CIDFP-Retrieve

This CID field pointer specifies where in call instance data the retrieved data element is to be written.

– CIDFP-Error

This CID field pointer specifies where in output call instance data the error cause will be written.

5.12.4.3 Call instance data

– Information value

Specifies the new value for the subscriber attribute data. This data is passed to this SIB from the user interaction SIB using the output CID parameter called collected data.

– Attribute indicator

Specifies the data attribute in the subscriber data object upon which the action is to be performed.

– Authorized Relationship Id

The Authorized Relationship Id provides the identity of the established Authorized Relationship through which the operations can be applied.

5.12.5 Output

5.12.5.1 Logical end

– Success (record written or retrieved);

– Error.

5.12.5.2 Call instance data

– Retrieved Value(s)

Specifies the data attribute(s) values retrieved.

– Error cause

Identifies the specific condition which caused an error during the operation of the SIB. The following errors have been identified for SERVICE DATA MANAGEMENT:

a) invalid object indicator;

b) invalid action;

c) invalid attribute indicator;

d) invalid information value;

e) invalid Inc/Dec value.

5.12.6 Graphical representation

See Figure 13.

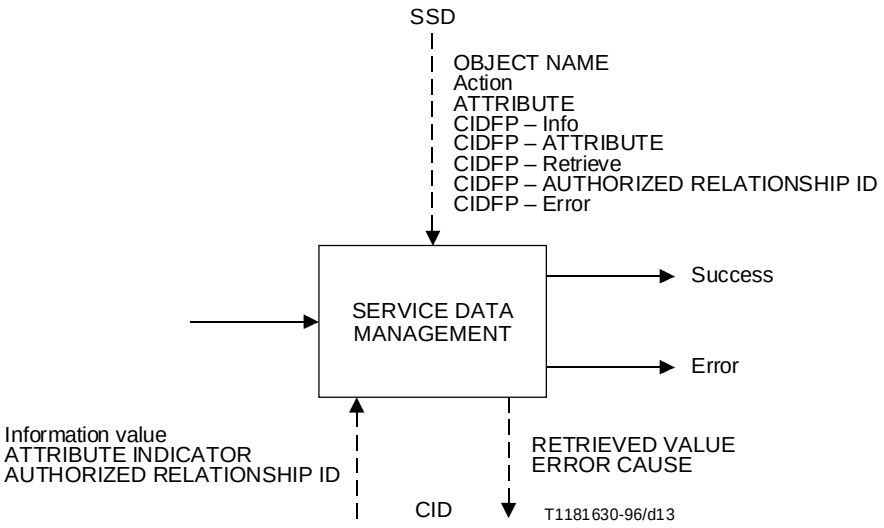

FIGURE 13/Q.1213

SERVICE DATA MANAGEMENT SIB

5.13 STATUS NOTIFICATION

5.13.1 Definition

Provide the capability of inquiring about the status and/or status changes of network resources.

5.13.2 Operation

There are four types of status notification requests:

– Poll resource status – Return the current status of the resource.

– Wait for status – Wait until the resource assumes the desired status unless the resource is already in the desired status.

– Initiate continuous monitor – Monitor and record the changes in busy/idle status; subsequent monitor – Return all the changes in status for the resource or other events as required.

– Cancel continuous monitor.

"Poll resource status" status notification could be used to determine if the destination address is busy or idle. In the busy case, "Wait for status" status notification could be used to notify service logic when the destination address becomes idle. "Initiate continuous monitor" status notification could be needed to create a resource history file.

5.13.3 Potential service applications

- CCBS;
- call distribution;
- freephone;
- call transfer.

5.13.4 Input

5.13.4.1 Logical start

Indicates the logical start of execution for the SIB.

5.13.4.2 Service support data

- Type

  Specifies the type of operation for this SIB. Four types have been identified:

  a) poll resource status;

  b) wait for status;

  c) initiate continuous monitor;

  d) cancel continuous monitor.

- Resource

  Specifies the particular entity to be monitored. The following resources can be monitored:

  1) lines;

  2) trunks.

- Timer

  Specifies the maximum amount of time to monitor the resource. If the value of timer is "null" no time limit is specified. This SSD is not used if the type is set to "poll resource status" or "cancel continuous monitor".

- Status notification file name

  Specifies the file object where the current resource status is to be logged. If a "null" value is specified, the resource status will not be logged. This SSD is not used if type is set to "cancel continuous monitor".

- Resource status

  Specifies the desired status of the resource being monitored. This SSD is only used when the type is set to "wait for status". Valid resource status's are:

  a) busy;

  b) idle.

- CIDFP-Status

  This CID field pointer specifies where in call instance data the resultant status of the resource is to be written.

- CIDFP-Error

  This CID field pointer specifies where in output call instance data the error cause will be written.

5.13.4.3 Call instance data

– None.

5.13.5 Output

5.13.5.1 Logical end

– Success

- Poll resource status – Present status returned.
- Wait for status – Line or resource has assumed the given status.
- Initiate continuous monitor – Continuous monitor initiated.
- Cancel continuous monitor – Continuous monitor terminated.

– Timer expiry (only used with "wait for status").

– Error.

5.13.5.2 Call instance data

– Status

Specifies the current status of the resource.

– Error cause

Identifies the specific condition which caused an error during the operation of the SIB. The following errors have been identified for STATUS NOTIFICATION:

a) invalid type;

b) invalid resource;

c) invalid timer;

d) invalid file indicator;

e) invalid resource status.

5.13.6 Graphical representation

See Figure 14.

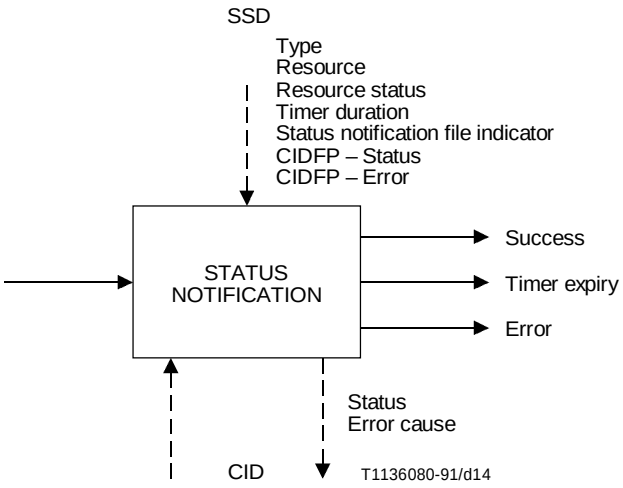

FIGURE 14/Q.1213

STATUS NOTIFICATION SIB

5.14 TRANSLATE

5.14.1 Definition

Determines output information from input information.

5.14.2 Operation

This SIB translates input information and provides output information, based on the various other input parameters. Parameters provided identify which file should be scanned for the translation. Translation can be based on either input information only, or on input information and the CLI.

For example, this SIB could be used for modifying input information (for instance, dialled digits) into a standard numbering plan upon which network routing is based.

In conjunction with other SIBs, like COMPARE, the TRANSLATE SIB can provide the required functionality for time dependant routing.

5.14.3 Potential service applications

– Freephone;

– User-defined routing;

– VPN;

– UPT;

– Abbreviated dialling;

– Selective call forwarding on busy/don't answer;

– Call forwarding;

– Call transfer.

5.14.4 Input

5.14.4.1 Logical start

Indicates the logical start of execution for the SIB.

5.14.4.2 Service support data

– Object Name

  Specifies where the translation data object is located.

– Translate Filter

  The Translate Filter identifies the attributes and filter tests to be applied to the attributes in the translate data object. The attribute values are stored in the CIDFP – Filter Value(s) pointers.

– Translated Attribute

  Specifies which object attribute(s) is to be returned to Translated data value(s).

– CIDFP-Filter Value(s)

  This CID field pointer(s) specifies which call instance data is to be used as the information.

– CIDFP-Translated Data

  This CID field pointer specifies where in call instance data that the translated data attribute value is to be written.

– CIDFP - Authorized Relationship Id

  This CID field pointer specifies which call instance data is to be used as the Authorized Relationship Id.

– CIDFP-Error

This CID field pointer specifies where in output call instance data the error cause will be written.

5.14.4.3 Call instance data

– Filter Value(s)

Specifies the translate filter data value(s) to be used. Stored in CIDFP - Filter Value(s).

– Authorized Relationship Id

The Authorized Relationship Id provides the identity of the established Authorized Relationship through which the operations can be applied.

5.14.5 Output

5.14.5.1 Logical end

– Success;

– Error.

5.14.5.2 Call instance data

– Translated data

Specifies the data element(s) resulting from the translation.

– Error cause

Identifies the specific condition which caused an error during the operation of the SIB. The following errors have been identified for TRANSLATE:

a) invalid type;

b) invalid file indicator;

c) invalid information;

d) translation not available.

5.14.6 Graphical representation

See Figure 15.

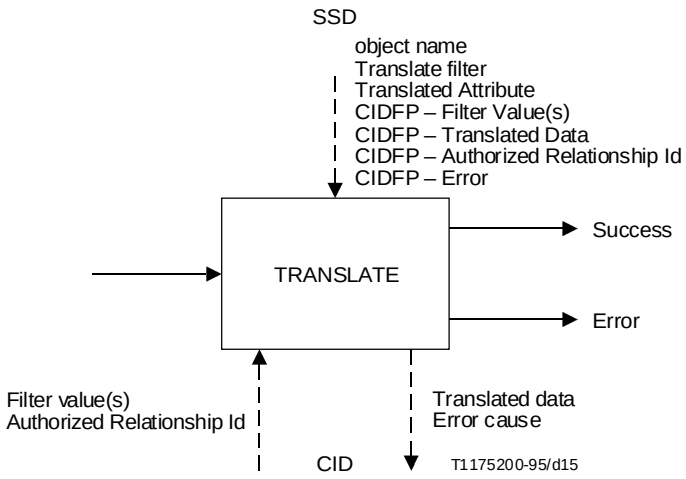

FIGURE 15/Q.1213

TRANSLATE SIB

5.15 USER INTERACTION

5.15.1 Definition

Allows information to be exchanged between the network and a call party, where a call party can be either a calling or a called party.

5.15.2 Operation

This SIB provides a call party with information (e.g. announcements) and/or collects information from a call party.

The announcements can be, for example:

- DTMF tones;
- a customized or generic audio message;
- network progression tones (e.g. dial tone, busy tone, etc.).

The collected information can be, for example:

- DTMF tones;
- audio;
- IA5 string text.

This SIB provides the specified announcement (which may be null) to the user. Depending on the repetition type (i.e. count, time), the message is repeated until the repetition number or time is reached.

Expected user input may be null. Assuming it is not null and user interruptibility is allowed, the message is stopped upon initial input by the user. If user interruptibility is not allowed or the message (or message sequence) is completed with no input from the user, the first input should be received within the initial response time after the message (or message sequence) is terminated, otherwise an error condition results.

User input is considered complete when the maximum number of characters has been reached, or an end delineator character is received, or the inter-digit timer has been exceeded.

The stage 2 SDL representation of this SIB may be helpful in understanding its operation. Refer to 5.2.12/Q.1214.

5.15.3 Potential service applications

- Most IN CS-1 services will require user interaction.

5.15.4 Input

5.15.4.1 Logical start

Indicates the logical start of execution for the SIB.

5.15.4.2 Service support data

- Announcement parameters

  Specify the following control values for announcements:

  a) Announcement ID

     Specifies which announcement is to be sent. The Announcement ID could be “null” to signify that no announcement is to be sent.

  b) Repetition requested

     Specifies if the announcement is to be repeated.

  c) Repetition interval

     Specifies the delay period in seconds between repetitions.

  d) Maximum repetitions

     Specifies the maximum number of times the announcement will be repeated.

e) Duration

Specifies the maximum length of time the announcement will be played.

– Collect information parameters

Specify the control values for user entered information.

1) User interruptibility

Specifies if an announcement can be interrupted by the call party entering information (yes or no).

2) Voice feedback

Specifies if the user is given a vocalization of the users input.

3) Type

Specifies the expected form of the user entered information. The following forms can be identified:

i) DTMF;

ii) audio;

iii) IA5 string;

iv) null (where null signifies that no data is to be collected).

4) Maximum number of characters

Specifies the maximum number of characters to collect, (> = 0, where 0 identifies non-character input).

5) Minimum number of characters

Specifies the minimum number of characters to collect, (< = the maximum, > = 0, where 0 identifies non-character input).

6) Initial input waiting timer

Specifies the maximum time to wait for beginning of call party's response.

7) Inter-character waiting timer

Specifies the maximum time to wait after a pause by the call party.

8) End delineator

Specifies special character(s) signifying the end of an input. If this parameter is "null", no delineator is specified.

– CIDFP-Call_Party

This CID field pointer specifies which call instance data is to be used to identify the call party.

– CIDFP-Collected

This CID field pointer specifies where in output call instance data that the call party entered data is to be written.

– CIDFP-Error

This CID field pointer specifies where in output call instance data the error cause will be written.

5.15.4.3 Call instance data

– Call party identifier

Specifies the data associated with the CIDFP-Call_Party which identifies the call party exchanging information with the network.

5.15.5 Output

5.15.5.1 Logical end

– Success;

– error.

5.15.5.2 Call instance data

– Collected data

Specifies the data collected by the network from the call party.

– Error cause

Identifies the specific condition which caused an error during the operation of the SIB. The following errors have been identified for USER INTERACTION:

a) call abandon;

b) collection time out (no input received);

c) incorrect number of digits received;

d) announcement resource unavailable;

e) data collection resource unavailable;

f) invalid announcement ID;

g) invalid call party;

h) inconsistent timer setting;

i) call status incompatible with playing of announcements;

j) call status incompatible with collecting information.

5.15.6 Graphical representation

See Figure 16.

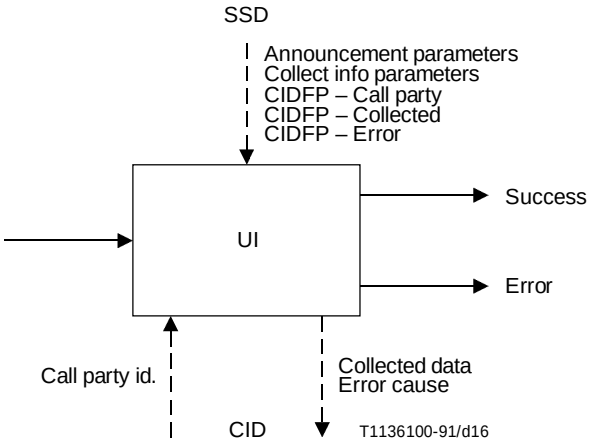

FIGURE 16/Q.1213

USER INTERACTION SIB

5.16 VERIFY

5.16.1 Definition

Provide confirmation that information received is syntactically consistent with the expected form of such information.

5.16.2 Operation

Information is compared with the format expected for the data. The VERIFY SIB normally follows the USER INTERACTION SIB when information has been collected from a call party.

The service support data identifies the type of data format. This SIB compares the input data to that expected format.

This SIB may, for example, be used when there is a need to verify the syntax of information. This could be:

- a user identification;
- a network address;
- any extra dialogue such as PIN code dialling or answer to a call prompt.

5.16.3 Potential service applications

- Freephone;
- selective call forwarding;
- UPT;
- televoting.

5.16.4 Input

5.16.4.1 Logical start

Indicates the logical start of execution for the SIB.

5.16.4.2 Service support data

- Maximum number of characters

  Specifies the maximum number of characters allowed.

- Minimum number of characters

  Specifies the minimum number of characters allowed.

- Format

  Format specifies the expected syntax of the data being verified. The format is specified using "code" characters. The following "codes" have been identified:

  1) x Any character;
  2) L Any letter;
  3) A Upper case letters only;
  4) a Lower case letters only;
  5) D Any digit (0-9) or delineator (#, *);
  6) N Any digit (0-9);
  7) n Any digit except 0 (1-9);
  8) [x] Optional character, where x represents any of the specified "codes";
  9) \x\ Specific character(s) required (e.g. \01\ indicated that either a 0 or a 1 must be present at that designated position in the data being verified).

     For example, the format "NNNN\c\" would indicate that a PIN code of 2387c would pass syntax verification, while 2387d would fail.

- CIDFP-Data

  This CID field pointer specifies which call instance data is to be used as the identifier to be verified.

- CIDFP-Error

  This CID field pointer specifies where in output call instance data the error cause will be written.

5.16.4.3 Call instance data

– Data

Specifies the data associated with the CIDFP-Data which is to be verified.

5.16.5 Output

5.16.5.1 Logical end

– Pass;

– Fail;

– Error.

5.16.5.2 Call instance data

– Error cause

Identifies the specific condition which caused an error during the operation of the SIB. The following errors have been identified for VERIFY:

a) invalid format;

b) invalid identifier;

c) inconsistent Max/Min number of characters specified.

5.16.6 Graphical representation

See Figure 17.

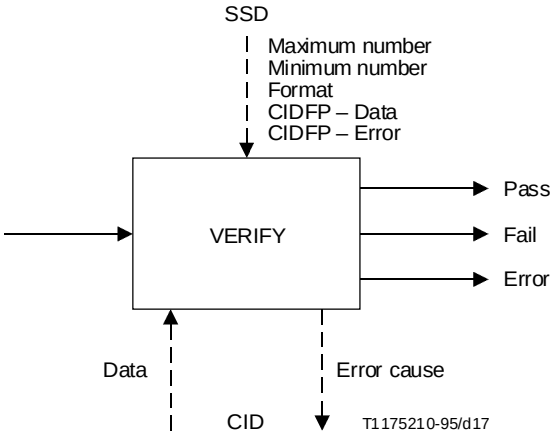

FIGURE 17/Q.1213

VERIFY SIB

6 Basic call process

6.1 General

The Basic Call Process (BCP) has been defined in 4.0/I.329/Q.1203 [2] as a specialized SIB which provides the basic call capabilities.

These capabilities enable the use of GSL as well as other SIBs to completely describe IN CS-1 services and service features.

It is not necessary or intended to fully model the BCP on the GFP in IN CS-1, but rather to insure that the functionality of the BCP that is required on the GFP in conjunction with SIBs be available to fully support IN CS-1 services/service features.

6.2 Points of initiation and points of return

The BCP functional interfaces to the global service logic in the GFP are described as Points of Initiation (POI) and Points of Return (POR). These are defined in 4.2/I.329/Q.1203.

IN CS-1 service flexibility is provided by defining specific POI and POR interfaces between the BCP and GSL. This flexibility allows a chain of SIBs to represent different services based on the launching POI and the returning POR.

For example, a SIB chain which screens a network address against a list could represent an originating call screening service if the address analysed POI launches the chain. However, if the prepared to complete call POI was used to launch the same chain, a terminating call screening service could result.

Furthermore, using the screening example from above, different services can result based upon which POR is used to complete the chain. If a "Match" logic end of the SCREEN SIB, is followed by a proceed with new data POR, a call is allowed to complete and this is an example of positive screening. However, negative screening, or call blocking, can be done with the same SIB chain by having the "Match" followed by a clear call POR.

6.2.1 List of POIs

The following set of POIs has been identified for IN CS-1.

Call originated

This POI identifies that the user has made a service request without yet specifying a destination address (e.g. off-hook but before dialling).

Address collected

This POI identifies that the address input has been received from the user.

Address analysed

This POI identifies that the address input has been analysed to determine characteristics of the address (e.g. freephone number).

Prepared to complete call

This POI identifies that the network is prepared to attempt completion of the call to the terminating party.

Busy

This POI identifies that the call is destined for a user who is currently busy.

No answer

This POI identifies that the call has been offered to a user who has not answered.

Call acceptance

This POI identifies that the call is active but the connection between the calling and called parties is not established (e.g. called party off-hook but no switch-through).

Active state

This POI identifies that the call is active and the connection between the calling and called parties is established.

End of call

This POI identifies that a call party has disconnected.

6.2.2 List of PORs

The following set of PORs has been identified for IN CS-1.

Continue with existing data

This POR identifies that the BCP should continue call processing with no modification.

Proceed with new data

This POR identifies that the BCP should proceed with call processing with only a data modification.

Handle as transit

This POR identifies that the BCP should treat the call as if it had just arrived.

Clear call

This POR identifies that the BCP should clear the call.

Enable call party handling

This POR identifies that the BCP should perform functions to enable call control for individual call parties.

The use of this POR and the definition of any additional SIBs needed to support call party handling, are for further study.

Initiate call

This POR identifies that the call should be initiated. This may be independent of an existing call, or may be in the context of an existing call.

6.3 BCP stage 1 description

6.3.1 Definition

This specialized SIB allows access to IN services/service features represented through the use of chains of SIBs and global service logic. The interface points between this SIB and GSL are described as POIs and PORs.

6.3.2 Operation

The BCP contains a set of armed POIs, and if during the processing of a call, one of these POIs is encountered, a chain of SIB is executed through global service logic.

When the chain of SIB terminates, call processing may be influenced according to the specified POR.

NOTE – Non-IN supported services are processed in the BCP and no GSL processing is required.

6.3.3 Potential service applications

– All IN CS-1 services.

6.3.4 Output

Specifies the POI and data parameters which are passed to GSL.

6.3.4.1 Logical output

Initiate global service logic from a specified POI.

6.3.4.2 Service support data

– Set of POIs

  Specifies the points in the BCP where IN service logic processing can occur for a given service.

– CIDFP-CLI

  This CID field pointer specifies which call instance data is to be used as the CLI.

– CIDFP-Category

  This CID field pointer specifies which call instance data is to be used as the CLI category data.

– CIDFP-Dialled

  This CID field pointer specifies which call instance data is to be used as the dialled number.

– CIDFP-Destination

  This CID field pointer specifies which call instance data is to be used as the destination number.

– CIDFP-Call reference

  This CID field pointer specifies which call instance data identifies the call reference.

– CIDFP-Bearer

  This CID field pointer specifies which call instance data is to be used as the CLI bearer capability.

6.3.4.3 Call instance data

– Calling line identity

  Specifies the network address from which the call in progress originated.

– Calling line category

  Specifies the characteristics of CLI (e.g. payphone, operator, etc.).

– Dialled number

  Specifies the number(s) dialled by the caller.

– Destination number

  Specifies the number(s) dialled by the caller (the destination number although originally the same as the dialled number, may be modified through IN service processing).

– Call reference

  Identifies a specific call.

– Bearer capabilities

  Specifies the ISDN bearer capabilities (see Recommendation Q.931 [7]) requested by the caller.

6.3.5 Input

6.3.5.1 Logical input

Resume BCP at the point specified by the GSL specified POR.

6.3.5.2 Call instance data

– Destination number

  Specifies a network address to which the call in progress is to be connected. For instance, a chain of SIBs may be used to create a destination number which differs from the dialled number.

6.3.6 Graphical representation

See Figure 18.

7 Global service logic

Global Service Logic (GSL) is the “glue” that describes the order in which SIBs can be chained together to accomplish services.

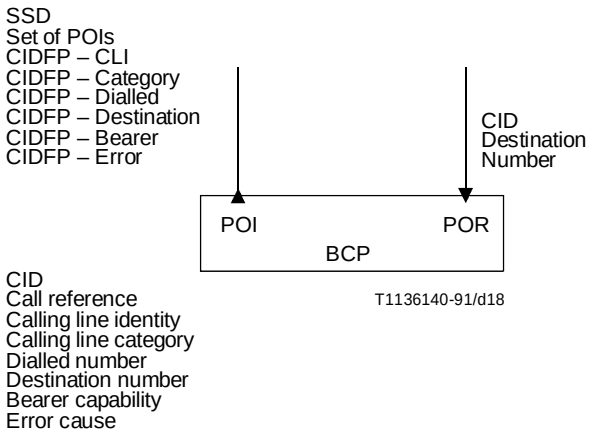

FIGURE 18/Q.1213

Basic call process SIB

For a given IN CS-1 services/service feature, global service logic is used to describe:

i) A specific POI which will define the functional launching point from the BCP to the SIB chain. Refer to 6.2.1 for a list of IN CS-1 POIs.

ii) A specific set of PORs where the SIB chain can logically return to the BCP. Refer to 6.2.2 for a list of IN CS-1 PORs.

iii) The pattern and order of SIBs which are to be chained together. This pattern begins at the POI defined in i) and ends at the set of PORs defined by ii). Refer to clause 5 for a list of IN CS-1 SIBs.

iv) Data parameters (SSD and CID) for each SIB in the SIB chain.

7.1 Relationship between GSL and BCP

Global service logic on the global functional plane views basic call process as a single resource. Based upon such a view of IN services, the following are identified as necessary interactions between global service logic and BCP, for example:

Communications from BCP to GSL:

i) *Logical start for SIB chains* – Which is represented by POIs.

ii) Data – Which is represented by call instance data, which is required by SIB chains for processing IN service features. Examples of specific call instance data which the BCP may be responsible for, could include calling line identity and dialled number.

Communications from GSL to BCP:

i) *Logical termination for SIB chains* – Which is represented by PORs.

ii) *Data* – Which is represented by call instance data that have been defined by one or more SIBs on a SIB chain. An example of such a call instance data could be a destination number. GSL ensures that all relevant CID is maintained throughout multiple SIB chains until termination of each call instance.

7.2 Relationship between global service logic and SIBs

The remaining components of GSL needed to define a service/SF, are the collection of SIBs (including their service support and call instance data), and the topology of their interconnection (to each other and to the POIs and PORs of the BCP). This specifies the functionality required to support the service/SF and the sequence of occurrence of this functionality.

8 Mapping of the service plane to the global functional plane

Referring to Figure 19 in the GFP, non-IN services are processed through the basic call process. When an IN Service Feature (SF) is to be evoked, it is initiated by a triggering mechanism from the basic call process. The chain "link" pattern which describes the SF, must then be obtained by the global service logic in order to process the SF. As new SFs are designed their SIB descriptions must be made available to the global service logic.

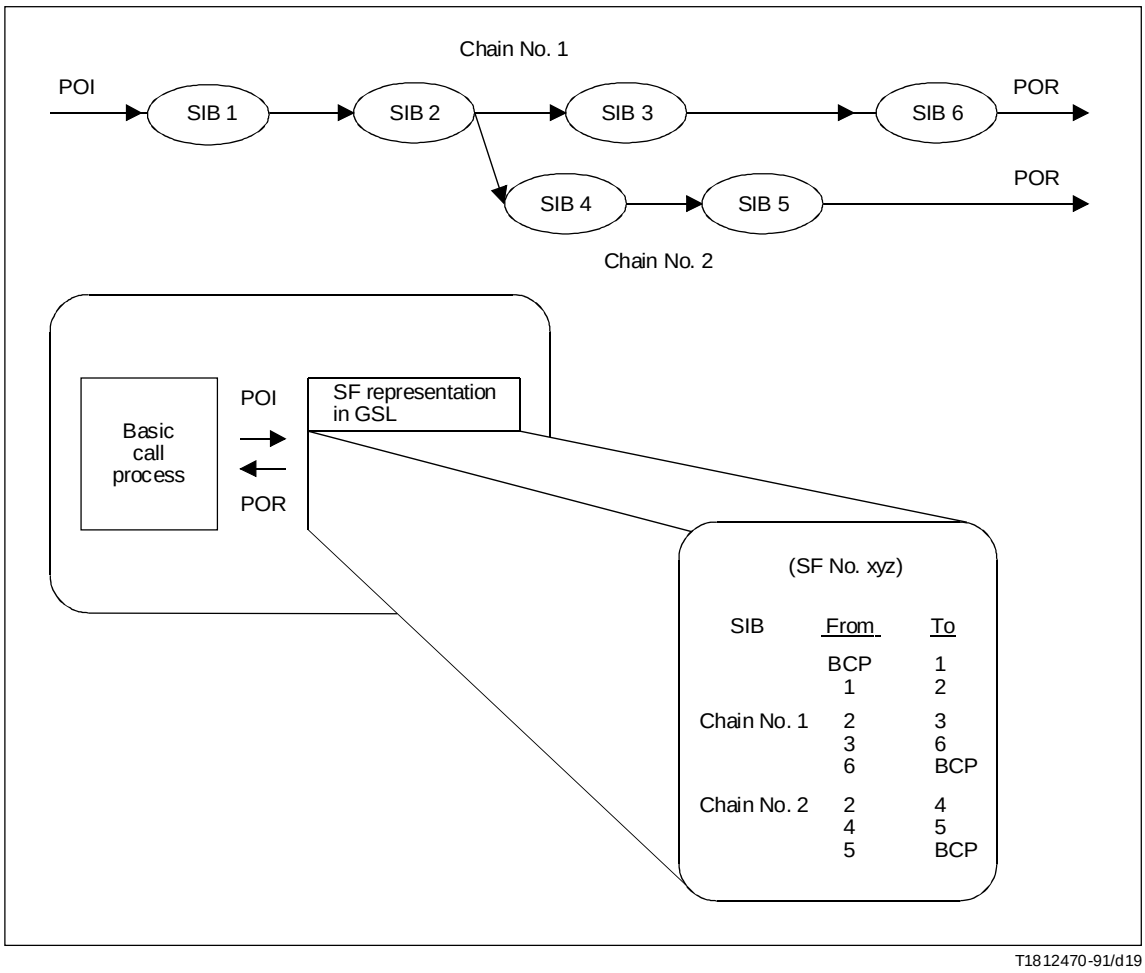

FIGURE 19/Q.1213

Global functional plane blueprint of a service feature