ETSI

ETSI TECHNICAL REPORT

ETR 321

December 1996

Source: ETSI TC-NA

Reference: DTR/NA-006005

ICS: 33.020

Key words: IN, CS1

Intelligent Network (IN); Global functional plane for IN Capability Set 1 (CS1)

ETSI

European Telecommunications Standards Institute

ETSI Secretariat

Postal address: F-06921 Sophia Antipolis CEDEX - FRANCE
Office address: 650 Route des Lucioles - Sophia Antipolis - Valbonne - FRANCE
X.400: c=fr, a=atlas, p=etsi, s=secretariat - **Internet:** secretariat@etsi.fr

Tel.: +33 4 92 94 42 00 - Fax: +33 4 93 65 47 16

Contents

Blank page

Foreword

This ETSI Technical Report (ETR) has been produced by the Network Aspects (NA) Technical Committee of the European Telecommunications Standards Institute (ETSI).

This ETR was previously published as the Technical Committee Reference Technical Report (TCR-TR)-016.

ETRs are informative documents resulting from ETSI studies which are not appropriate for European Telecommunication Standard (ETS) or Interim European Telecommunication Standard (I-ETS) status. An ETR may be used to publish material which is either of an informative nature, relating to the use or the application of ETSs or I-ETSs, or which is immature and not yet suitable for formal adoption as an ETS or an I-ETS.

Blank page

I Scope

The scope of this ETSI Technical Report (ETR) is the further development of the Intelligent Network (IN) for both the Integrated Services Digital Network (ISDN) and Public Switched Telecommunication Network (PSTN).

This ETR is based on ITU-T Recommendation Q.1213 [1] as given in CCITT COM XI-R 164, 1992. The requirements of this Recommendation apply, unless modified by the statements provided in Clause III of this ETR.

This ETR is intended as a guide to implementors and network operators on IN Capability Set 1 (CS1) Service Independent building Blocks (SIBs).

II References

This ETR incorporates by dated or undated reference, provisions from other publications. These references are cited at the appropriate places in the text and the publications are listed below. For dated references subsequent amendments to, or revisions of, any of these publications apply to this ETR only when incorporated in it by amendment or revision. For undated references the latest edition of the publication referred to applies.

[1] ITU-T Recommendation Q.1213: "Global Functional Plane for IN CS-1".

[2] ETS 300 374: "Core Intelligent Network Application Protocol (INAP)".

III Exceptions to ITU-T Recommendation Q.1213

In the following clauses the entire ITU-T Recommendation Q.1213 [1] is reproduced with additional text underlined and deleted text struck out. To minimize the confusion, the paragraphs in these clauses have the same numbering as ITU-T Recommendation Q.1213 [1].

1 General

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

1.1 Scope of IN Global Functional Plane for Capability Set 1

IN Capability Set 1 (CS1) 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 CSl. General GFP aspects are addressed in associated Recommendations I.312/Q.1201 and I.329/Q.1203.

The following functional characteristics are specific to CS1 and are addressed in this Recommendation:

- CS1 SIBs (Section 2.0 of this Recommendation)
  For CS1, twenty Service Independent Building Blocks (SlBs) are specified, (not including the BCP SIB discussed below). These represent the minimum set of SIBs required to define the CS1 targeted services, identified in associated Recommendation Q.1211.

- BCP SIB (Section 3.0 of this Recommendation)
  For CS1 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, two Points of Return (PORs) for GSL interfaces have been specified.

- Plane to Plane Mapping (Section 5.0 of this Recommendation)
  The relationships between the Service Plane and the GFP are specified for CS1.

- For CS1, 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 CS1 SIBs only where the SDLs clarify the understanding of the SIB operation.

1.2 Role of SlBs in the Global Functional Plane

SlBs are abstract representations of network capabilities that exist in an IN structured network. Their definition enforces the concept of service and technology independence by de coupling the services from the technology on which services are provided. The twenty SlBs defined in this Recommendation identify the network capabilities which will be available to service designers in a CS1 IN structured network. Their use will provide valuable insight on how service creation will be facilitated in CSn Recommendations.

The actual service implementation may, however, need an enhanced set of SIBs. The implementation aspects are outside the scope of this ETR.

1.3 CS1 Global Functional Plane model

The INCM described in Recommendation I.312/Q.1201 models the GFP as shown in figure 1/Q.1213. 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 CS1 POIs and PORs, the CS1 SlBs, 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 CS1 service features in the GFP.

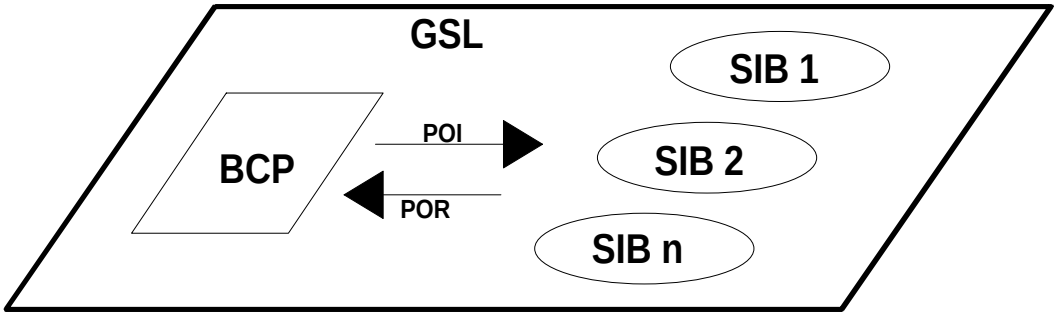

Figure 1/Q.1213: Global Functional Model

1.4 Terminology

The following terminology used in this Recommendation, is more fully defined in Recommendations I.329/Q.1203 and Q.1290:

CID	Call Instance Data
CIDFP	Call Instance Data Field Pointer
CLI	Calling Line Identification
SSD	Service Support Data

2 CS1 Service Independent Building Blocks (SIBs)

The following list of SIBs have been identified as required to support the list of targeted CS1 services and service features identified in Recommendation Q.1211:

Algorithm
Charge
Compare
Distribution
Limit
Log Call Information
Queue
Screen
Service Data Management
Status Notification
Translate
User Interaction
Verify

The following SIBs have been added by ETSI:

Connect
Continue
Disconnect Resource
Edp Info
Edp Request
Initiate Call
Release Call

The Stage 1 SIB descriptions that follow reflect the understanding of the logical function of each SIB in its role of supporting CS1 services and service features. The BCP, which is viewed as a specialized SIB, is described in Section 3.0.

An explanation of the format of the Stage 1 SIB descriptions can be found in Recommendation I.329/Q.1203.

The datatypes of data involved in the Service Independent Building Blocks (CID/SSD) are specified. A list of datatypes is included in section 6.0.

2.1 Algorithm

2.1.1 Definition

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

2.1.2 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.

2.1.3 Potential Service Applications

- Mass Calling
- Televoting

2.1.4 Input

2.1.4.1 Logical Start

Indicates the logical start of execution for the SIB.

2.1.4.2 Service Support Data

- Type: AlgorithmActionType
  Specifies the type of algorithm for this SIB. Two algorithms have been identified for CS1:
  A Increment
  B Decrement

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

- CIDFP - Algorithm: PointerType
  This CID Field Pointer specifies the Call Instance Data to which the algorithm is to be applied.

- CIDFP - Error: PointerType
  This Call Field Pointer specifies where in output Call Instance Data the error cause will be written.

2.1.4.3 Call Instance Data

- Identifier: IntegerType
  The Identifier is the data associated with the CIDFP-Algorithm upon which algorithm is to be applied.

2.1.5 Output

2.1.5.1 Logical End

- Success
- Error

2.1.5.2 Call Instance Data

- Identifier: IntegerType
  The Identifier is the data associated with the CIDFP-Algorithm upon which algorithm is to be applied. It contains the resultant value after the SIB has completed.

- Error Cause: ErrorType
  Identifies the specific condition which caused an error during the operation of the SIB. The following errors have been identified for Algorithm:
  - Invalid Type
  - Invalid Value

2.1.6 Graphical Representation

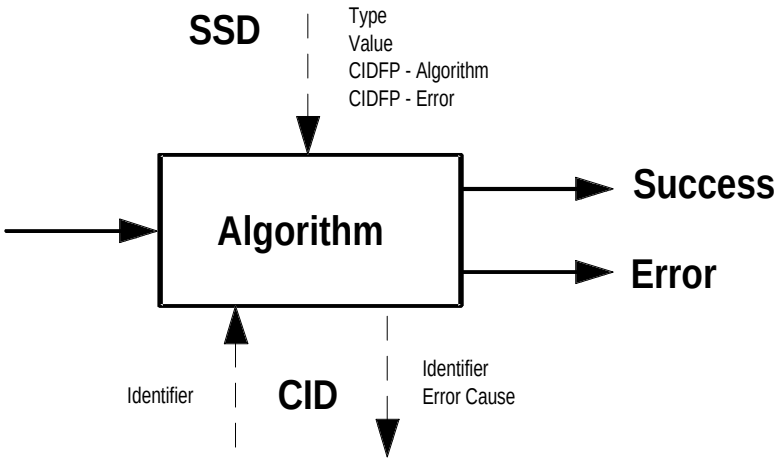

Figure 2/Q.1213: Algorithm

2.2 Charge

2.2.1 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.

2.2.2 Operation

The charging information output by this SIB shall 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 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 to 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:

- the account identified by the CLI,
- the account identified by the called number (either the dialled number or the destination number).
- an account or credit card identified by the collected user information,
- the calling users exchange accumulator,
- a pay phone.

2.2.3 Potential Service Applications

Any service which requires specific IN charging.

2.2.4 Input

2.2.4.1 Logical Start

Indicates the logical start of execution for the SIB.

2.2.4.2 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:
  - CIDFP-Line
    This CID Field Pointer specifies which Call Instance Data is to be used as the line number to charge.
  - CIDFP-Account
    This CID Field Pointer specifies which Call Instance Data is to be used as the account number to charge.
  - Fixed account
    Specifies an account number which is fixed for all call instances.

  B Percent *(%)*
  Specifies the allocation of the total charge for this account. The sum of all allocations shall 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.

2.2.4.3 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.

2.2.5 Output

2.2.5.1 Logical End

1 Success
2 Error

2.2.5.2 Call Instance Data

- Error Cause: ErrorType
  Identifies the specific condition which caused an error during the operation of the SIB. The following errors have been identified for Charge:
  - Invalid account to charge
  - Invalid percentage allocation (> 100, < 0)
  - Invalid sum of percentage allocations (not equal to 100)
  - Invalid service/service feature
  - Invalid resource
  - Invalid units
  - System fault - unable to write record

2.2.6 Graphical Representation

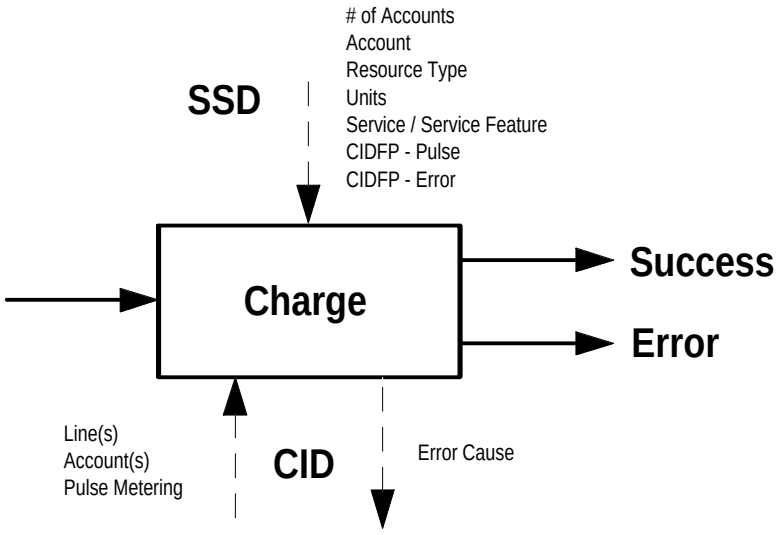

Figure 3/Q.1213: Charge SIB

2.3 Compare

2.3.1 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.

2.3.2 Operation

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

2.3.3 Potential Service Applications

- Time Dependent Routing
- CCBS

2.3.4 Input

2.3.4.1 Logical Start

Indicates the logical start of execution for the SIB.

2.3.4.2 Service Support Data

- CIDFP - Compare: PointerType
  This CID Field Pointer specifies which Call Instance Data is to be used as the Identifier.

- Reference Value: CompType
  Specifies the value against which the comparison will be made.

- CIDFP - Error: PointerType
  This CID Field Pointer specifies where in output Call Instance Data the error cause will be written.

2.3.4.3 Call Instance Data

- Identifier: CompType
  Specifies the data associated with the CIDFP-Compare (See SSD) which is to be compared to the Reference Value.

2.3.5 Output

2.3.5.1 Logical End

- GREATER THAN the value
- LESS THAN the value
- EQUAL TO the value
- ERROR

2.3.5.2 Call Instance Data

- Error Cause: ErrorType
  Identifies the specific condition which caused an error during the operation of the SIB.
  The following errors have been identified for Compare:
  - Invalid Identifier
  - Invalid Reference Value

2.3.6 Graphical Representation

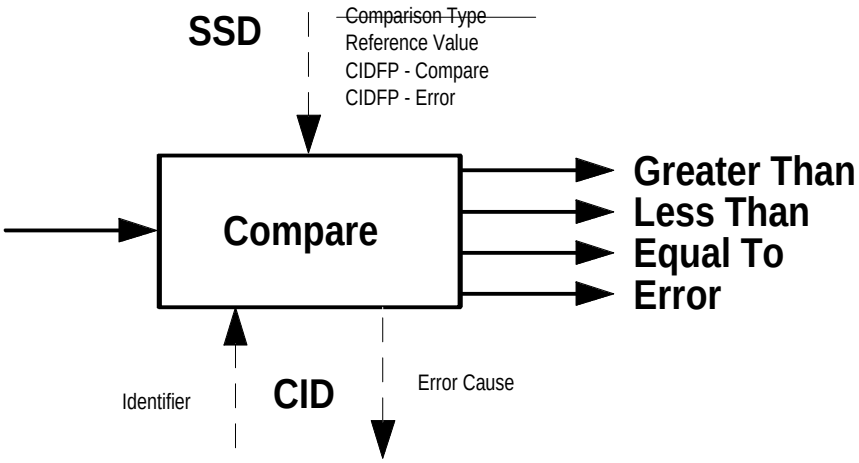

Figure 4/Q.1213: Compare SIB

2.4 Distribution

2.4.1 Definition

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

2.4.2 Operation

This SIB distribute 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

- Algorithm Type
  A Percentage
  B Sequential
  C Time Of Day
  D Day Of Week

  NOTE: Hierarchical call distribution may be realized through the use of multiple instances of the Distribution SIB in conjunction with the Status Notification SIB.

2.4.3 Potential Service Applications

- Mass Calling
- Televoting
- Freephone

2.4.4 Input

2.4.4.1 Logical Start

Indicates the logical start of execution for the SIB.

2.4.4.2 Service Support Data

- Number of Logical Ends: IntegerType

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

- Algorithm Parameters: DistributionAlgorithmType

- CIDFP - Error: ErrorType
  This CID Field Pointer specifies where in output Call Instance Data the error cause will be written.

2.4.4.3 Call Instance Data

- None.

2.4.5 Output

2.4.5.1 Logical End

- 1
- 2

...

- N (where N = Number of Logical Ends (item 1 of Input SSD)
- Error

2.4.5.2 Call Instance Data

- Error Cause: ErrorType
  Identifies the specific condition which caused an error during the operation of the SIB.
  The following errors have been identified for Distribution:
  - Invalid Type
  - Invalid percentage allocation (> I 00, < 0)
  - Invalid sum of percentage allocations (not equal to 100)
  - Invalid Number of Logical Ends
  - Missing time period
  - Overlap of the time across logical ends
  - Missing days
  - Overlap of days across logical ends

2.4.6 Graphical Representation

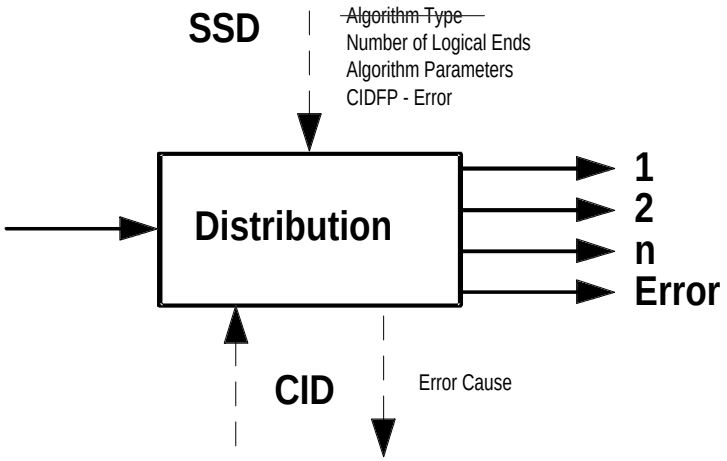

Figure 5/Q.1213: Distribution SIB

2.5 Limit

2.5.1 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.

2.5.2 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).

2.5.3 Potential Service Applications

- Mass Calling
- Televoting
- Freephone

2.5.4 Input

2.5.4.1 Logical Start

Indicates the logical start of execution for the SIB.

2.5.4.2 Service Support Data

- Parameters: LimType
- CIDFP - File: PointerType
  This CID Field Pointer specifies where the current Limit count is located.

- CIDFP - Error: PointerType
  This CID Field Pointer specifies where in output Call Instance Data the error cause will be written.

2.5.4.3 Call Instance Data

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

2.5.5 Output

2.5.5.1 Logical End

- Pass
- No pass
- Error

2.5.5.2 Call Instance Data

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

- Error Cause: ErrorType
  Identifies the specific conditions which caused an error during the operation of the SIB.
  The following errors have been identified for Limit:
  - Invalid Type
  - Invalid Parameters

2.5.6 Graphical Representation[N2]

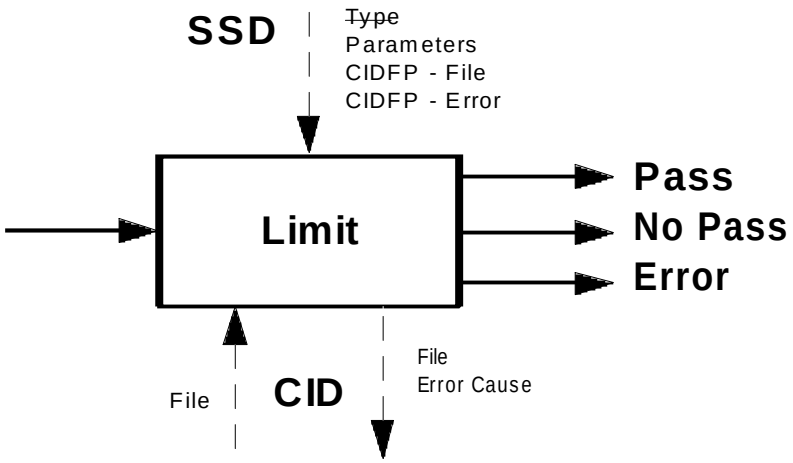

Figure 6/Q.1213: Limit SIB

2.6 Log Call Information

2.6.1 Definition

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

2.6.2 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.

2.6.3 Potential Service Application

- All services

2.6.4 Input

2.6.4.1 Logical Start

Indicates the logical start of execution for the SIB.

2.6.4.2 Service Support Data

- CIDFP-Log
  These CID Field Pointers specify which Call Instance Data are to be used as Identifiers.
  CIDFP-Log can include:
  - call attempt time,
  - call stop time,

- call connect time,
- destination number,
- additional dialled number (e.g., credit card number, etc.),
- calling line identification,
- time in queue,
- bearer capability,
- error causes,
- any other CID.

- Log File Indicator
  Specifies the log file where the value of the Identifier is to be logged.

- CIDFP - Error
  This CID Field Pointer specifies where in output Call Instance Data the error cause will be written.

2.6.4.3 Call Instance Data

- Identifier(s)
  Specifies the data associated with the CIDFP -Log (See SSD) which is to be logged.

2.6.5 Output

2.6.5.1 Logical End

- Success (information is requested)
- Error

2.6.5.2 Call Instance Data

- Error Cause: ErrorType
  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

2.6.6 Graphical Representation

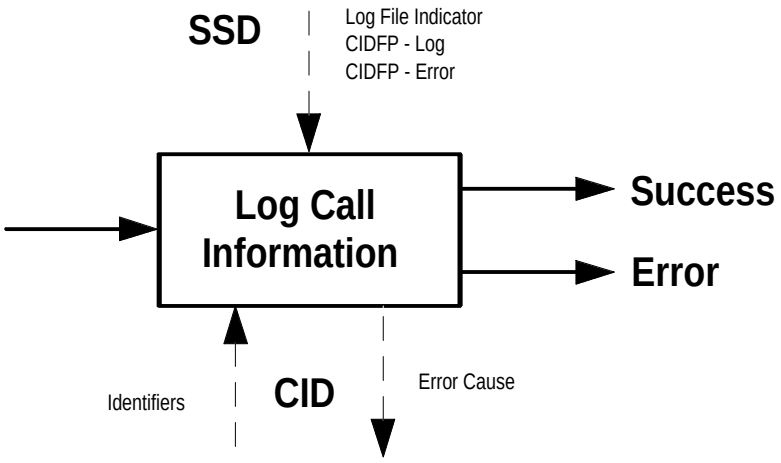

Figure 7/Q.1213: Log Call Information SIB

2.7 Queue

2.7.1 Definition

Provide sequencing of IN calls to be completed to a called party [1].

2.7.2 Operation

This SIB provides all the processing needed to provide queuing for a call, and will specifically:
- pass the call if resources are available,
- queue the call,
- play announcements to caller on queue,
- when resources become available, de-queue the call.

2.7.3 Potential Service Applications

All services which use Call Queuing service feature.

2.7.4 Input

2.7.4.1 Logical Start

Indicates the logical start of execution for the SIB.

2.7.4.2 Service Support Data

- Max active: IntegerType
  Specifies the maximum number of active calls allowed for the resource.

- Max number: IntegerType
  Specifies the maximum number of calls allowed on queue at a given time.

- Max time: ApplicationTimerType
  Specifies the maximum time the call may remain on the queue.
- Announcement Parameters: AnncInfoType
  Specify the control values for announcements. The control values which can be specified are:
  - Announcement ID specifies which announcement is to be sent,
  - Repetition Interval specifies the play period in seconds between repetitions,
  - Maximum Repetitions specifies the maximum number of times the announcement will be repeated.
- CIDFP-Resource: PointerType
  This CID Field Pointer specifies which Call Instance Data identifies the Resource.

- CIDFP-Error: PointerType
  This CID Field Pointer specifies where in output Call Instance Data the error cause will be written.

2.7.4.3 Call Instance Data

- Call Reference: CallReferenceType
  Identifies the specific call which is a candidate for queuing.

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

---

[1] The current definition of Queue SIB is not flexible enough to meet the requirements from all CS1 "Benchmark Services" e.g., UPT

2.7.5 Output

2.7.5.1 Logical End

1 Resource available
2 Call party abandon
3 Queue timer expire
4 Queue full
5 Error

2.7.5.2 Call Instance Data

- Time spent in queue: ApplicationTimerType
  Identifies the total time that a particular call was queued.

- Error Cause: ErrorType
  Identifies the specific condition which caused an error during the operation of the SIB.
  The following errors have been identified for Queue:
  - Invalid Max active
  - Invalid Max number
  - Invalid Max time
  - Invalid Announcement Parameters
  - Invalid Call Reference

2.7.6 Graphical Representation

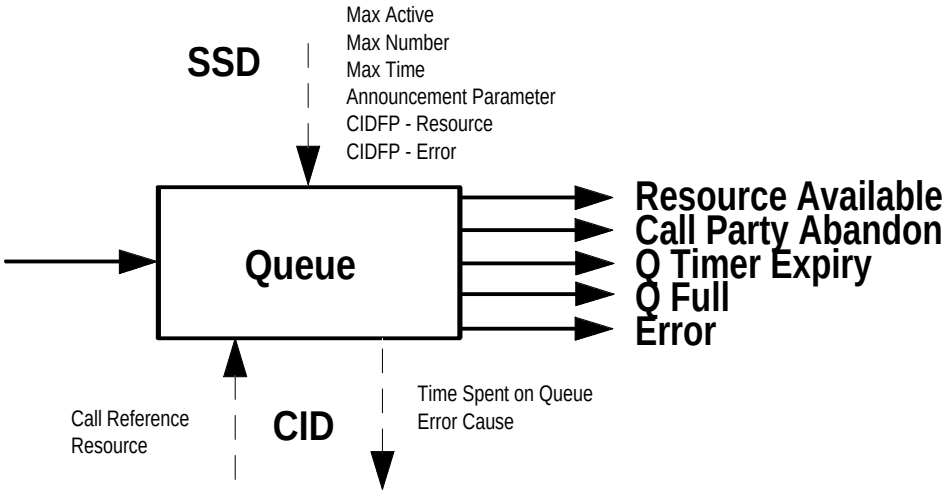

Figure 8/Q.1213: Queue SIB

2.7.7 SDL Diagram

2.8 Screen

2.8.1 Definition

Perform a comparison of an identifier against a list to determine whether the identifier has been found in the list [2].

2.8.2 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 Translation and Compare, may result in more complex screening capabilities.

2.8.3 Potential Service Applications

- Selective Call Forward on Busy/Don't Answer
- Originating Call Screening
- Terminating Call Screening
- Security Screening
- Account Card Calling
- Credit Call Calling

2.8.4 Input

2.8.4.1 Logical Start

Indicates the logical start of execution far the SIB.

2.8.4.2 Service Support Data

- Screen List Indicator: ScreenListIndType
  The Screen List Indicator identifies the screen data to be used.

- CIDFP - Screen: PointerType
  This CID Field Pointer specifies which Call Instance Data is to be used as the Identifier.

- CID - Error: PointerType
  This CID Field Pointer specifies where in output Call Instance Data the error cause will be written.

2.8.4.3 Call Instance Data

- Identifier: ScreenDataType
  The Identifier is the data associated with the CIDFP-Screen (See SSD) which is to be screened against the screen data.

2.8.5 Output

2.8.5.1 Logical End

- Match [=on the list]
- No Match

---

[2] The current definition of Screen SIB is not flexible enough to meet the requirements from all CS1 "Benchmark Services" e.g., UPT

- Error

2.8.5.2 Call Instance Data

- Error Cause: ErrorType
  Identifies the specific condition which caused an error during the operation of the SIB.
  The following errors have been identified for Screen:
  - Invalid Identifier
  - Invalid Screen List

2.8.6 Graphical Representation

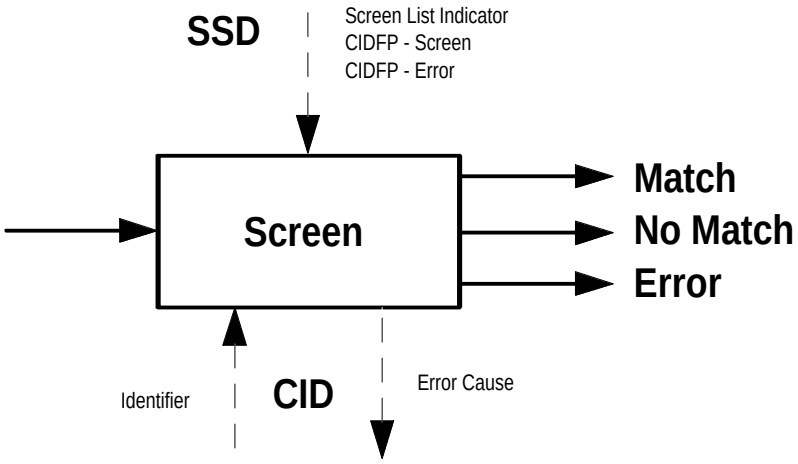

Figure 9/Q.1213: Screen SIB

2.9 Service Data Management

2.9.1 Definition

Enables end user specific data to be replaced, retrieved, incremented, or decremented[3].

2.9.2 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 customers call forwarding number.

2.9.3 Potential Service Applications

- Call Forwarding
- Customer Profile Management

2.9.4 Input

2.9.4.1 Logical Start

Indicates the logical start of execution for the SIB.

---

3 The current definition of SDM SIB is not flexible enough to meet the requirements from all CS1 "Benchmark Services" e.g., UPT

2.9.4.2 Service Support Data

- File indicator: DataBaseType
  Specifies the subscriber data file to be used.

- Action: SdmActionType
  Specifies the operation to be performed on the subscriber data. The following Actions are allowed:

  Replace - Replace the existing data element in the subscriber file specified by the Element Indicator with new data specified by the Information Value.
  Retrieve - Copy the data element specified by the Element Indicator and place it in the Output CID called Data Retrieved.
  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.
  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.

- Element Indicator: ElementType
  Specifies the data element in the subscriber data file 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 Element Indicator can vary and its value shall be provided through Call Instance Data.

- Inc/Dec Value: IntegerType
  Specifies the amount by which the Element Indicator is to be incremented or decremented.

- CIDFP - Info: PointerType
  This CID Field Pointer identifies which Call Instance Data is to be used as the Information Value.

- CIDFP - Element: PointerType
  This CID Field Pointer specifies which Call Instance Data is to be used as the Element Indicator. If the Element Indicator is to be specified by the SSD, then this CID Field Pointer will not be used by the SIB.

- CIDFP - Retrieve: PointerType
  This CID Field Pointer specifies when in Call Instance Data the retrieved data element is to be written.

- CIDFP - Error: PointerType
  This CID Field Pointer specifies where in output Call Instance Data the error cause will be written.

2.9.4.3 Call Instance Data

- Information Value: ANY
  Specifies the new value for the subscriber data. This data is passed to this SIB from the User Interaction SIB using the output CID parameter called Collected Data

- Element Indicator: ElementType
  Specifies the data element in the subscriber data file upon which the Action is to be performed. This optional data parameter is only required when multiple data elements in the subscriber data file can be changed (e.g., in the case of an Abbreviated Dialling List).

2.9.5 Output

2.9.5.1 Logical End

- Success (record written or retrieved)
- Error

2.9.5.2 Call Instance Data

- Data Retrieved: ANY
  Specifies the data element retrieved

- Error Cause: ErrorType
  Identifies the specific condition which caused an error during the operation of the SIB.
  The following errors have been identified for Service Data Management:
  - Invalid File Indicator
  - Invalid Action
  - Invalid Element Indicator
  - Invalid Information Value
  - Invalid Inc/Dec Value

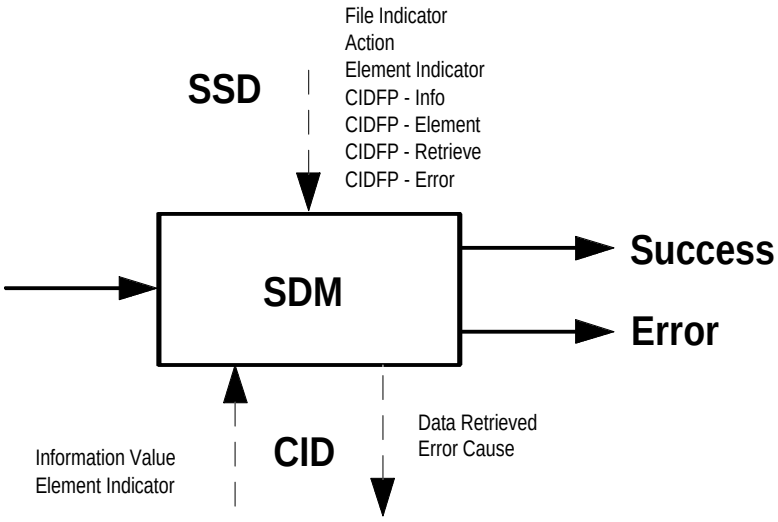

Figure 10/Q.1213: Service Data Management SIB

2.10 Status Notification

2.10.1 Definition

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

2.10.2 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.
- 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.

2.10.3 Potential Service Applications

- CCBS
- Call Distribution
- Freephone
- Call Transfer

2.10.4 Input

2.10.4.1 Logical Start

Indicates the logical start of execution for the SIB.

2.10.4.2 Service Support Data

- Type: StatusNotificationActionType
  Specifies the type of operation for this SIB. Four types have been identified:
  - Poll resource status
  - Wait for status
  - Initiate continuous monitor
  - Cancel continuous monitor

- Resource: ResourceType
  Specifies the particular entity to be monitored. The following resources can be monitored:
  - Lines
  - Trunks

- Timer: IntegerType
  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 Indicator: FileIdType
  Specifies the file 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".

  Specifies the desired status of the resource being monitored This SSD is only used when the Type is set to "Wait far status". Valid Resource Status are:
  - Busy
  - Idle

- CIDFP - Status: PointerType
  This CID Field Pointer specifies where in Call Instance Data that the resultant status of the resource is to be written.

- CIDFP - Error: PointerType
  This CID Field Pointer specifies where in output Call Instance Data the error cause will be written.

2.10.4.3 Call Instance Data

- none.

2.10.5 Output

2.10.5.1 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 Expire (only used with "Wait for status")

- Error

2.10.5.2 Call Instance Data

- Status: StatusType
  Specifies the current status of the resource.

- Error Cause: ErrorType
  Identifies the specific conditions which caused an error during the operation of the SIB.
  The following errors have been identified for Status Notification:
  - Invalid Type
  - Invalid Resource
  - Invalid Timer
  - Invalid File Indicator
  - Invalid Resource Status

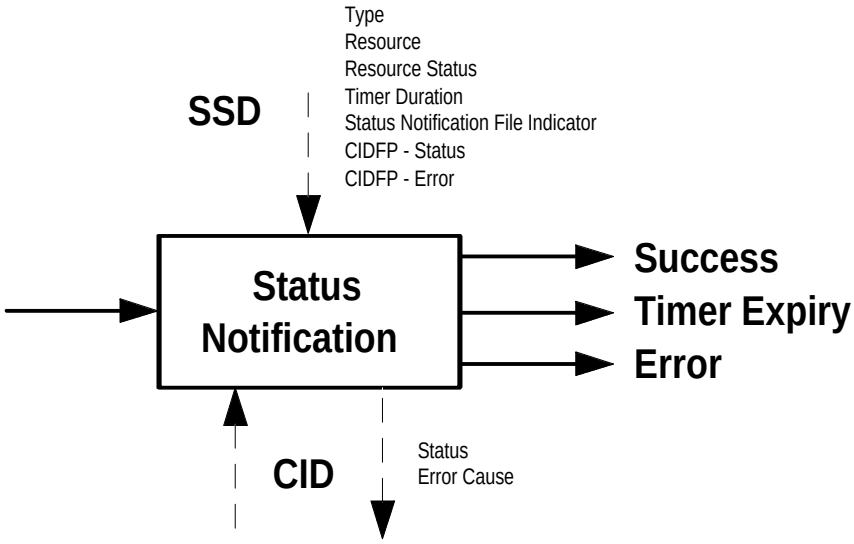

Figure 11/Q.1213: Status Notification SIB

2.11 Translate

2.11.1 Definition

Determines output information from input information.

2.11.2 Operation[N3]

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 Translation Key.

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.

2.11.3 Potential Service Applications

- Freephone
- User-defined routing
- VPN
- UPT
- Abbreviated dialling
- Selective Call Forwarding on Busy/Don't Answer
- Call Forwarding
- Call Transfer

2.11.4 Input

2.11.4.1 Logical Start

Indicates the logical start of execution for the SIB.

2.11.4.2 Service Support Data

- Type: TranslateType
  Specifies the mode of operation for this SIB. Three modes have been identified:
  - One number to one number
  - One number to more than one number
  - IA5 string to one number

- File Indicator: DataBaseType
  Specifies when the translation data file is located.

- CIDFP - Translation Key: PointerType
  This CID Field Pointer specifies the Translation Key e.g., Calling Line Identification.

- CIDFP - Info: PointerType
  This CID Field Pointer specifies which Call Instance Data is to be used as the Information.

- CIDFP - Translated: PointerType
  This CID Field Pointer specifies where in Call Instance Data that the translated data element is to be written.

- CIDFP - Error: PointerType
  This CID Field Pointer specifies where in output Call Instance Data the error cause will be written.

2.11.4.3 Call Instance Data

- Translation Key: DigitsType
  Specifies the Translation Key

- Information: DigitsType
  Specifies the data to be translated.

2.11.5 Output

2.11.5.1 Logical End

- Success
- Error

2.11.5.2 Call Instance Data

- Translated data: SET ANY
  Specifies the data element(s) resulting from the translation.

- Error Cause: ErrorType
  Identifies the specific condition which caused an error during the operation of the SIB.
  The following have been identified for Translate:
  - Invalid Type
  - Invalid File Indicator
  - Invalid Information
  - Invalid Translation Key
  - Translation not available

2.11.6 Graphical Representation[N5]

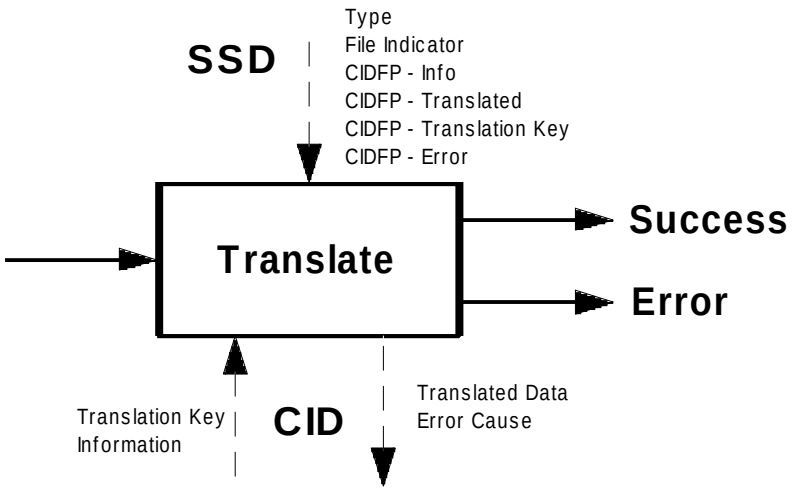

Figure 12/Q.1213: Translate SIB

2.12 User Interaction

2.12.1 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.

2.12.2 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 Section 5.2.12 of associated Recommendation Q.1214.

2.12.3 Potential Service Applications

Most CS1 services will require User Interaction.

2.12.4 Input

2.12.4.1 Logical Start

Indicates the logical start of execution for the SIB.

2.12.4.2 Service Support Data

- Announcement Parameters: AnncInfoType
  Specify the following control values for announcements:
  - Announcement ID
    Specifies which announcement is to be sent. The Announcement ID could be "null" to signify that no announcement is to be sent.

  Maximum Repetitions
    Specifies the maximum number of times the announcement will be repeated.
  - Repetition Interval
    Specifies the delay period in seconds between repetitions.

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

- Collect Information Parameters: CollectType
  Specify the control values for user entered information.

  - User interruptibility
    Specifies if an announcement can be interrupted by the call party entering information(Yes or No).
  - Voice feedback
    Specifies if the user is given a vocalisation of the users input.
  - Type
    Specifies the expected form of the user entered information. The following forms can be identified:
      DTMF,
      Audio,
      IA5 String,
      Null (where Null signifies that no data is to be collected)
  - Maximum number of characters
    Specifies the maximum number of characters to collect, (>=0, where 0 identifies non-character input).
  - Minimum number of characters
    Specifies the minimum number of characters to collect, (<= the minimum, >• , where 0 identifies non-character input).
  - Initial Input waiting timer
    Specifies the maximum time to wait for beginning of call party's response.
  - Inter-character waiting timer
    Specifies the maximum time to wait or a pause by the call party.
  - End delineator
    Specifies special character(s) signifying the end of an input. If this parameter is "null", no delineator is specified.

- CIDFP - Call Party: PointerType
  This CID Field Pointer specifies which Call Instance Data is to be used to identify the call party.

- CIDFP - Collected: PointerType
  This CID Field Pointer specifies where in output Call Instance Data that the call party entered data is to be written.

- CIDFP - Error: PointerType
  This CID Field Pointer specifies where in output Call Instance Data the error cause will be written.

2.12.4.3 Call Instance Data

- Call Party Identifier: LegIdType
  Specifies the data associated with the CIDFP - Call_Party which identified the call party exchanging information with the network.

2.12.5 Output

2.12.5.1 Logical End

- Success
- Error

2.12.5.2 Call Instance Data

- Collected data: CollectType
  Specifies the data collected by the network from the call party.

- Error Cause: ErrorType
  Identifies the specific condition which caused an error during the operation of the SIB.
  The following errors have been identified for User Interaction:
  - Call abandon
  - Collection Time out (No input:received.)
  - Incorrect number of digits received
  - Announcement resource unavailable
  - Data collection resource unavailable
  - Invalid Announcement
  - Invalid Call Party
  - Inconsistent timer setting
  - Call status incompatible with playing of announcements
  - Call status incompatible with collecting information

2.12.6 Graphical Representation[N7]

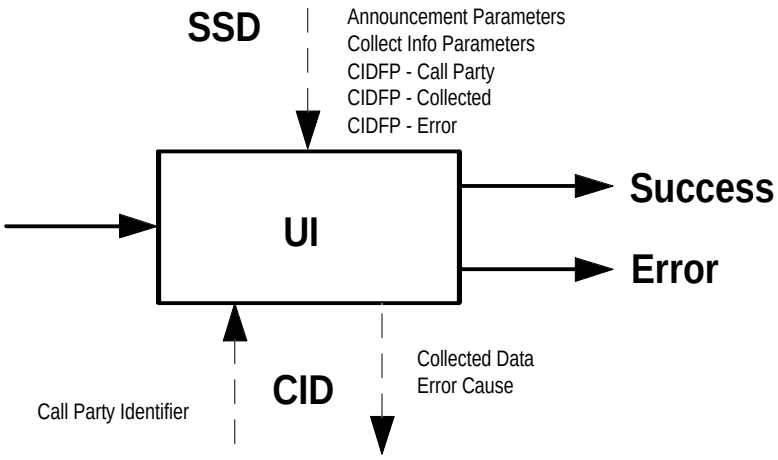

Figure 13/Q.1213: User Interaction SIB

2.13 Verify

2.13.1 Definition

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

2.13.2 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.

2.13.3 Potential Service Applications

- Freephone
- Selective Call Forwarding
- UPT
- Televoting

2.13.4 Input

2.13.4.1 Logical Start

Indicates the logical start of execution for the SIB.

2.13.4.2 Service Support Data

- Maximum number of characters: IntegerType
  Specifies the maximum number of characters allowed.

- Minimum number of characters: IntegerType
  Specifies the minimum number of characters allowed.

- Format: FormatType
  Format specifies the expected syntax of the data being verified. The format is specified using "code" characters. The following "codes" have been identified:
  x Any character
  L Any letter
  A Upper case letter only
  a Lower case letter only
  D Any digit (0-9) or delineator (#, *)
  N Any digit (0-9)
  n Any digit except 0 (1-9)
  [x] Optional character, where x represents any of the specified "codes"
  \x\ Specific character(s) required. (e.g. \O1\ indicated that either a 0 or a 1 shall 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.

  This CID Field Pointer specifies which Call Instance Data is to be used as the Identifier to be verified.

- CIDFP - Error: PointerType
  This CID Field Pointer specifies where in output Call Instance Data the error cause will be written.

2.13.4.3 Call Instance Data

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

2.13.5 Output

2.13.5.1 Logical End

- Pass
- Fail
- Error

2.13.5.2 Call Instance Data

- Error Cause: ErrorType
  Identifies the specific condition which caused an error during the operation of the SIB.
  The following errors have been identified for Verify:
  - Invalid Format
  - Invalid Identifier
  - Inconsistent Max/Min number of characters specified.

2.13.6 Graphical Representation

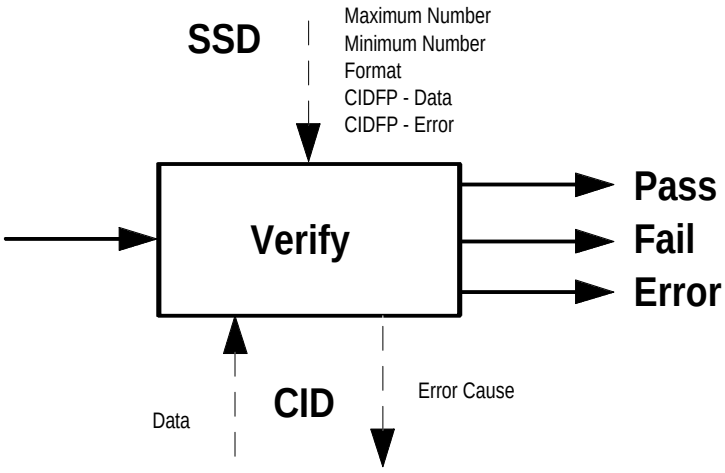

Figure 14/Q.1213: Verify SIB

2.14 Connect

2.14.1 Definition

Completes a call to a defined destination.

2.14.2 Operation

The Connect SIB is involved in the context of an existing call. It connects the calling party to the party specified by the connect parameters. After the invocation (or the processing) of this SIB the BCP resumes with new data.

2.14.3 Potential Service Applications

All CS1 services.

2.14.4 Input

2.14.4.1 Logical Start

Indicates the logical start of execution for the SIB.

2.14.4.2 Service Support Data

- CIDFP - Destination: PointerType
  This CID Field Pointer specifies which Call Instance Data is to be used as the Destination Number.

- CIDFP - CLI: PointerType
  This CID Field Pointer specifies which Call Instance Data is to be used as the Calling Line Identification.

- CIDFP - Bearer: PointerType
  This CID Field Pointer specifies which Call Instance Data is to be used as the CLI Bearer Capabilities.

- CIDFP - Category: PointerType
  This CID Field Pointer specifies which Call Instance Data is to be used as the CLI Category data.

- CIDFP - Error: PointerType
  This CID Field Pointer specifies where in output Call Instance Data the error cause will be written.

2.14.4.3 Call Instance Data

- Destination Number: DigitsType
  Specifies a network address to which the call in progress is to be connected.

- CLI: DigitsType
  Specifies the network address from which the call in progress originated.

- Bearer Capability: DigitsType
  Specifies the ISDN bearer capabilities (Q931) requested by the caller.

- Calling Line Category: DigitsType
  Specifies the characteristics of the CLI (e.g., pay phone, operator, etc.).

2.14.5 Output

2.14.5.1 Logical End

- Success
  Connect submitted to BCP.

- Error
  Error occurred during the operation of the SIB.

2.14.5.2 Call Instance Data

- Error Cause: ErrorType
  Identifies the specific condition which caused an error during the operation of the SIB. The following errors have been identified for Connect:
  - Invalid Destination Number
  - Invalid CLI

2.14.6 Graphical Representation

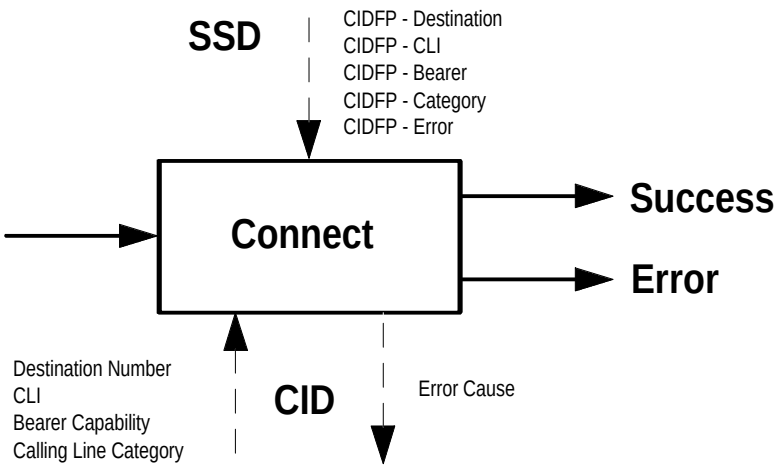

Figure 15/Q.1213: Connect SIB

2.15 Continue

2.15.1 Definition

Continues basic call processing from the point at which it was suspended.

2.15.2 Operation

This SIB can be used to indicate to the BCP to proceed call processing without data modifications. After the invocation (or the processing) of this SIB the BCP resumes with existing data.

2.15.3 Potential Service Applications

All CS1 services.

2.15.4 Input

2.15.4.1 Logical Start

Indicates the logical start of execution for the SIB.

2.15.4.2 Service Support Data

- none.

2.15.4.3 Call Instance Data

- none.

2.15.5 Output

2.15.5.1 Logical End

- Success
  Continue submitted to BCP.

2.15.5.2 Call Instance Data

- none.

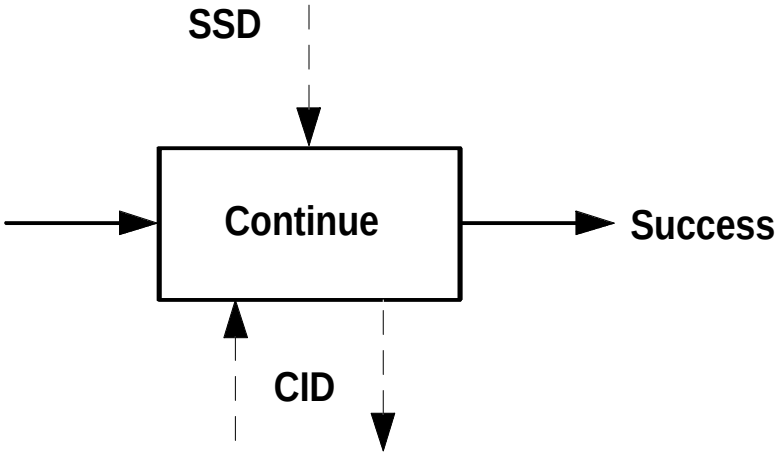

Figure 16/Q.1213: Continue SIB

2.16 Disconnect Resource

2.16.1 Definition

Release all specialized resources.

2.16.2 Operation

This SIB instructs the BCP to disconnect the specialized resources involved in the call.

2.16.3 Potential Service Applications

Most CS1 services will require the Disconnect Resource.

2.16.4 Input

2.16.4.1 Logical Start

Indicates the logical start of execution for the SIB.

2.16.4.2 Service Support Data

- none.

2.16.4.3 Call Instance Data

- none.

2.16.5 Output

2.16.5.1 Logical End

- Success
  Request for disconnection submitted to BCP.

2.16.5.2 Call Instance Data

- none.

2.16.6 Graphical Representation

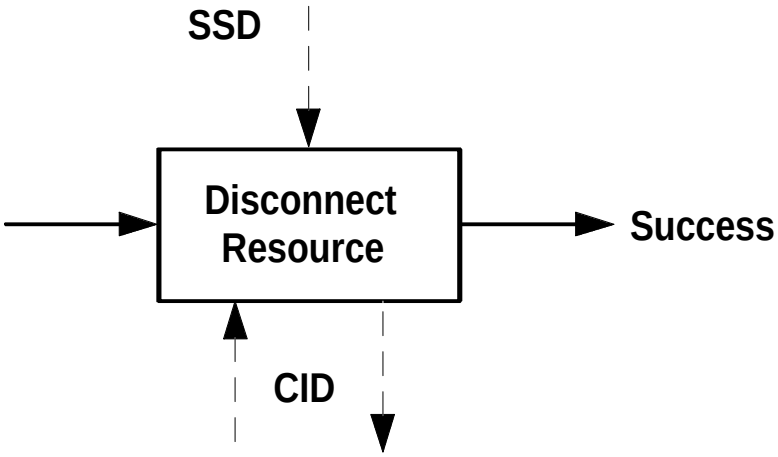

Figure 17/Q.1213: Disconnect Resource

2.17 Edp Info

2.17.1 Definition

Performs the retrieval of EDP information.

2.17.2 Operation

This SIB performs the actual retrieval of the EDP information, once an event has arrived.

2.17.3 Potential Service Applications

Most CS1 services.

2.17.4 Input

2.17.4.1 Logical Start

Indicates the logical start of execution for the SIB.

2.17.4.2 Service Support Data

- CIDFP - Specific: PointerType
  This CID Field Pointer specifies which Call Instance Data is to be used as the event specific information.

- CIDFP - Call_Party: PointerType
  This CID Field Pointer specifies which Call Instance Data is to be used as the call party.

- CIDFP - Message: PointerType
  This CID Field Pointer specifies which Call Instance Data is to be used as the message type.

2.17.4.3 Call Instance Data

- none.

2.17.5 Output

2.17.5.1 Logical End

- Success:
  EDP information retrieved, EDP cleared.

2.17.5.2 Call Instance Data

- Event Specific Information: ANY
  Specifies, if any, the information specific to the considered event.

- Call Party Identifier: LegType
  Specifies the data which identifies the call party to which the request is related.

- Message Type: IdentifierType
  Specifies the message type, which can be either "requested" or "modification".

2.17.6 Graphical Representation[N9]

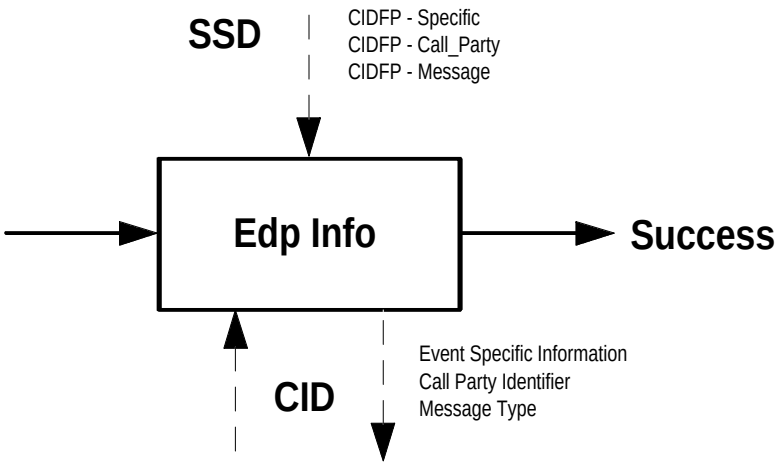

Figure 18/Q.1213: Edp Info

2.18 Edp Request

2.18.1 Definition

Arms Event Detection Points (EDPs) in the basic call process.

2.18.2 Operation

This SIB can be used to arm event detection points (EDPs) in the BCP. The requested event will occur after a POR.

2.18.3 Potential Service Applications

Most CS1 services.

2.18.4 Input

2.18.4.1 Logical Start

Indicates the logical start of execution for the SIB.

2.18.4.2 Service Support Data

- EDPs: EDPType
  Specifies the set of EDPs to arm.

  NOTE: If EDP = O_No_Answer or if EDP = T_No_Answer

  - timer (specifies the time after which the No_Answer result is established).
    If EDP = CollectedInfo
  - number of digits (specifies the number of digits to be collected).

- CIDFP - Error: PointerType
  This CID Field Pointer specifies where in output Call Instance Data the error cause will be written.

2.18.4.3 Call Instance Data

- none.

2.18.5 Output

2.18.5.1 Logical End

- Success
  EDP arm request submitted.

- Error
  An error occurred during EDP arming.

2.18.5.2 Call Instance Data

- Error Cause: ErrorType
  Identifies the specific condition which caused an error during the operation of the SIB. The following errors have been identified for Edp Request:
  - At least one EDP is impossible to arm.

2.18.6 Graphical Representation

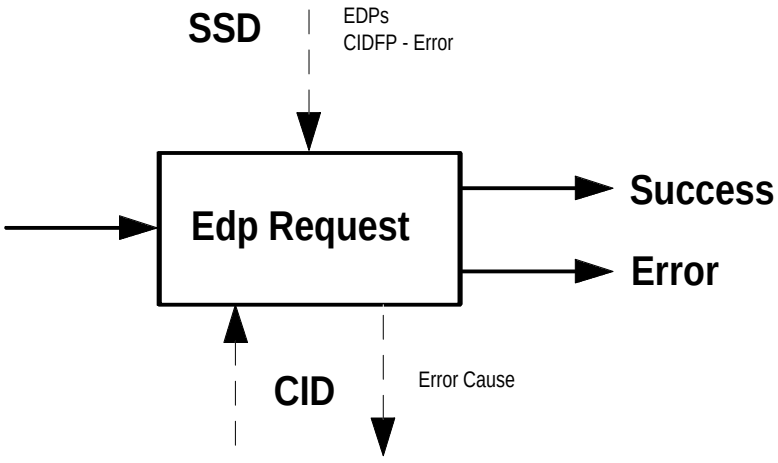

Figure 19/Q.1213: Edp Request

2.19 Initiate Call

2.19.1 Definition

Initiate a call to a defined destination.

2.19.2 Operation

The Initiate Call SIB is invoked while no relationship with the BCP exists.
A control relationship is established; this is also referred to as a service logic initiate call. After the invocation (or the processing) of this SIB the BCP is invoked.

2.19.3 Potential Service Applications

All CS1 services.

2.19.4 Input

2.19.4.1 Logical Start

Indicates the logical start of execution for the SIB.

2.19.4.2 Service Support Data[N11]

- Destination Number: DigitsType
  Specifies a network address to which the call in progress is to be connected.

- CLI: DigitsType
  Specifies the network address from which the call in progress originated.

- Bearer Capability: DigitsType
  Specifies the ISDN bearer capabilities (Q931) requested by the caller.

- Calling Line Category: DigitsType
  Specifies the characteristics of the CLI (e.g., pay phone, operator, etc.).

- CIDFP - Destination: PointerType
  This CID Field Pointer specifies which Call Instance Data is to be written as the Destination Number.

- CIDFP - CLI: PointerType
  This CID Field Pointer specifies which Call Instance Data is to be written as the Calling Line Identification.

- CIDFP - Bearer: PointerType
  This CID Field Pointer specifies which Call Instance Data is to be written as the CLI Bearer Capabilities.

- CIDFP - Category: PointerType
  This CID Field Pointer specifies which Call Instance Data is to be written as the CLI Category data.

- CIDFP - Error: PointerType
  This CID Field Pointer specifies where in output Call Instance Data the error cause will be written.

2.19.4.3 Call Instance Data

- none.

2.19.5 Output

2.19.5.1 Logical End

- Success
  Connect submitted to BCP.

- Error
  Error occurred during the operation of the SIB.

2.19.5.2 Call Instance Data

- Destination Number: DigitsType
  Specifies a network address to which the call in progress is to be connected.

- CLI: DigitsType
  Specifies the network address from which the call in progress originated.

- Bearer Capability: DigitsType
  Specifies the ISDN bearer capabilities (Q931) requested by the caller.

- Calling Line Category: DigitsType
  Specifies the characteristics of the CLI (e.g., pay phone, operator, etc.).

- Error Cause: ErrorType
  Identifies the specific condition which caused an error during the operation of the SIB. The following errors have been identified for Initiate Call:
  - Invalid destination number
  - Invalid CLI
  - Invalid Category

2.19.6 Graphical Representation[N12]

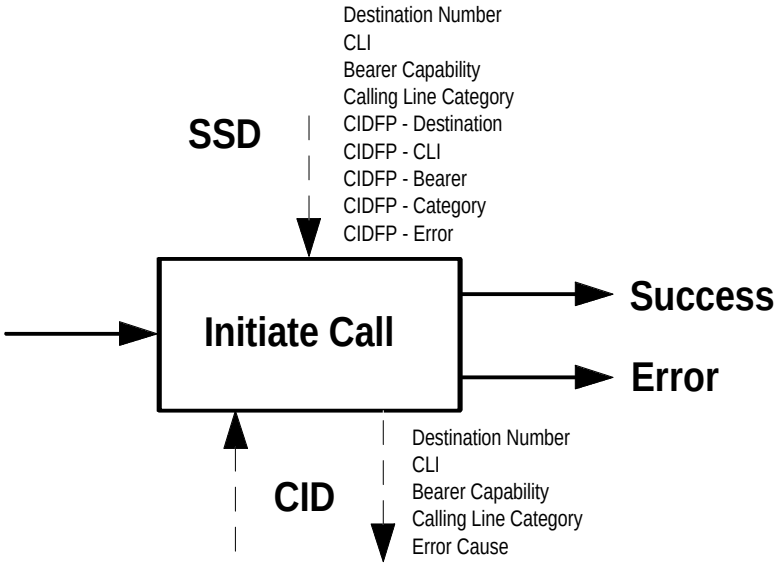

Figure 20/Q.1213: Initiate Call SIB

2.20 Release Call

2.20.1 Definition

Releases the call during any phase of call processing.

2.20.2 Operation

This SIB has to be used to terminate a call and continue service logic processing. After the invocation (or the processing) of this SIB the BCP will be cleared.

2.20.3 Potential Service Applications

All CS1 services.

2.20.4 Input

2.20.4.1 Logical Start

Indicates the logical start of execution for the SIB.

2.20.4.2 Service Support Data

- Cause: IdentifierType
  Specifies the reason for releasing the call.

2.20.4.3 Call Instance Data

- none.

2.20.5 Output

2.20.5.1 Logical End

- Success
  ReleaseCall submitted to BCP.

2.20.5.2 Call Instance Data

- none.

2.20.6 Graphical Representation

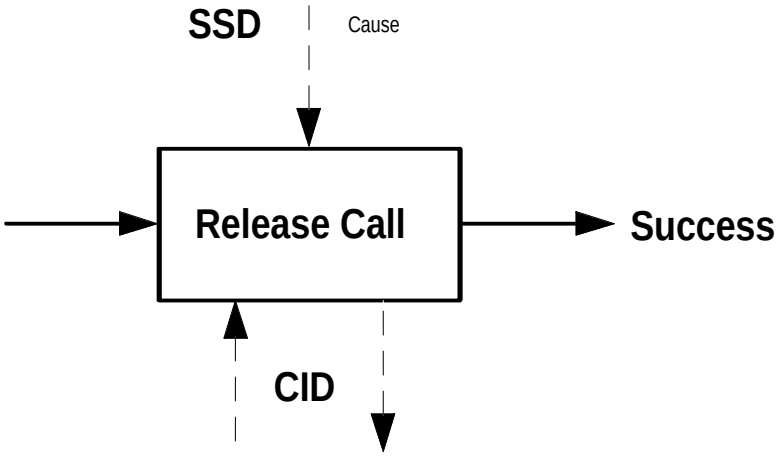

Figure 21/Q.1213: Release Call SIB

3 Basic Call Process

3.1 General

The Basic Call Process (BCP) has been defined in section 4.0 of Recommendation I.329/Q.1203 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 CS1 services and service features.

It is not necessary or intended to fully model the BCP on the GFP in CS1, 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 CS1 services/service features.

3.2 Points Of Initiation and Points Of Return[N13]

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 section 4.2 of Recommendation I.329/Q.1203.

CS1 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 CONNECT SIB, 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 RELEASE CALL SIB.

3.2.1 List of POIs

The following set of POIs has been identified for CS1:

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 far 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.

3.2.2 List of PORs

The following set of PORs has been identified for CS1:

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

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 is for further study).

3.3 BCP Stage 1 description

3.3.1 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.

3.3.2 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 that non-IN supported services are processed in the BCP and no GSL processing is required.

3.3.3 Potential Service Applications

All CS1 services.

3.3.4 Output

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

3.3.4.1 Logical output

Initiate Global Service Logic from a specified POI.

3.3.4.2 Service Support Data

- Set of POIs
  Specifies the points in the BCP where IN service logic processing can occur for a given services.

- 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.

- 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.

3.3.4.3 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., pay phone, 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 (Q.931) requested by the caller.

3.3.5 Input

3.3.5.1 Logical input

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

3.3.5.2 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.

3.3.6 Graphical Representation

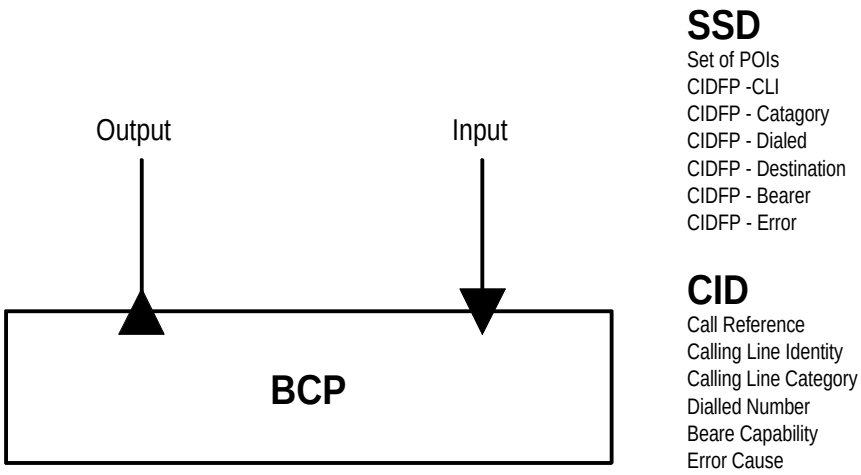

Figure 22/Q.1213: Basic Call Process SIB

4 Global Service Logic

Global Service Logic (GSL) has been defined in Section 5.0 of associated Recommendation I.329/Q.1213 as the "glue" that describes the order in which SIBs can be chained together to accomplish services.

For a given CS1 service/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 Section 3.2.1 for a list of CS1 POIs).

ii) A specific set of PORs where the SIB chain can logically return to the BCP (Refer to Section 3.2.2 for a list of CS1 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 Section 2.0 for a list of CS1 SIBs).

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

5 Mapping of the Service Plane to the Global Functional Plane

No unique CS1 Service Plane to Global Functional Plane mapping relationships have been identified over and above what has been defined in I.329/Q.1203.

```
ActionType ::=
        ENUMERATED {
                increment (0),
                decrement (1),
                replace (3),
                retrieve (4),
                pollResourceStatus (5),
                waitForStatus (6),
                intitiateContinuousMonitor (7),
                cancelContinuousMonitor (8)
        }

AlgorithmActionType ::= ActionType (increment | decrement)

AnncInfoType ::=
        SEQUENCE {
                announcementId          ANY         -- see Core INAP MessageType
                -- specifies which announcement is sent

                repetitionInterval      INTEGER,
                -- specifies the delay period in seconds between repetitions

                maximumRepetitions          INTEGER
                -- specifies the maximum number of times the announcement will be repeated
        }

ApplicationTimerType ::= ApplicationTimer                    -- see Core INAP

CallReferenceType ::=   OCTET STRING

CollectType ::=         OCTET STRING

CompType ::=
        CHOICE {
                identifierValue     INTEGER,
                timeOfDay           TimeOfDayType,
                dayOfWeek DayOfWeekType,
                dayOfYear           DayOfYearType
        }
```

```
CountCallType ::=
        SEQUENCE {
                N       INTEGER                 -- number of calls
                p       INTEGER                 -- counter interval
                -- pass (n) calls out of (p) calls.
        }

CountTimeType ::=
        SEQUENCE {
                s       ApplicationTimerType    -- duration in seconds
                q       ApplicationTimerType    -- interval in seconds
                -- pass calls for (s) seconds out of every (q) seconds.
        }

DataBaseType ::= DatabaseID             -- see Core INAP

DayOfWeekArrayType ::= SEQUENCE (SIZE (1..N)) OF DaysOfWeekType
        -- N represents the number of logical ends.
        -- All days of the week shall be accounted for. No day may be associated with more then one logical
        end.

DayOfWeekType ::=
        ENUMERATED {
                sun (0),
                mon (1),
                tue (2),
                wed (3),
                thu (4),
                fri (5),
                sat (6)
        }

DaysOfWeekType ::= SET OF DayOfWeekType

DigitsType ::=                  -- see Q.763 address signal digits

DistributionAlgorithmType ::=
        CHOICE {
                percentage  PercentageArrayType,
                sequential  NULL,
                timeOfDay       TimeOfDayArrayType,
                dayOfWeek DayOfWeekArrayType
        }

EDPType ::=
        SEQUENCE {
                event       EventType,
                monitor             MonitorType,
                leg         LegIdType,
                applTimer           TimerType
        }

ElementType ::=
        CHOICE {
                SEQUENCE {
                        key     ANY,            -- key to access the data
                        elementId   ANY         -- id of the element to operate on
                }
                NULL
        }

ErrorType ::=           -- To be specified e.g. INTEGER or PrintableString
```

```
EventType ::= EventTypeBCSM          -- see Core INAP

FileIdType ::=
        -- e.g.
        CHOICE {
                NULL,                -- file is not found
                SEQUENCE {
                        dataBaseId  DataBaseType
                        element     ElementType
                }
        }

FormatType ::=      PrintableString

IdentifierType ::=  OCTET STRING

IntegerType ::= INTEGER

LegIdType ::= LegId                  -- see Core INAP

LimType ::=
        CHOICE {
                interval        CountTimeType,
                numberOfCalls   CountCallType
        }

MonitorType ::= MonitorMode          -- see Core INAP

PercentageArrayType ::=         SEQUENCE (SIZE (1..N)) OF PercentageType
        -- N represents the number of logical ends.
        -- Sum over all array members shall equal 100.

PercentageType ::= INTEGER (1..100)

PointerType ::=             OCTET STRING

ResourceIDType ::= ResourceID                -- see Core INAP

ResourceType ::= ResourceID          -- see Core INAP

ScreenDataType ::= SET OF OCTET STRINGS

ScreenListIndType ::=
        -- Identifies the screen data list to use.
        SEQUENCE {
                databaseId  DataBaseType,
                extensions  SEQUENCE SIZE (0..MAX) OF ExensionField OPTIONAL
                -- extensions (to be defined)
        }

SdmActionType ::= ActionType (replace | retrieve | increment | decrement)

StatusNotificationActionType ::=            ActionType (pollResourceStatus | waitForStatus |
                                            initiateContinuousMonitor | cancelContinuousMonitor)

StatusType ::= ResourceStatus        -- see Core INAP

TimeOfDayArrayType ::= SEQUENCE (SIZE (1..n)) OF TimeOfDayIntervalType
        -- N represents the number of logical end.
        -- All 24 hours of the day shall be accounted for, with no overlap across logical ends.

TimeOfDayIntervalType ::=
```

```
        SEQUENCE {
                start   TimeOfDayType,
                end     TimeOfDayType
        }

TimeOfDayType ::= UTCTime

TranslateType ::=
        ENUMERATED {
                oneNumberToOneNumber (0),
                oneNumberToMoreThenOneNumber (1),
                iA5StringToOneNumber (2)
        }
```

History

Document history	
December 1996	First Edition

ISBN 2-7437-1151-5
Dépôt légal : Decembre 1996