INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Q.1223

(09/97)

SERIES Q: SWITCHING AND SIGNALLING

Intelligent Network

Global functional plane for intelligent network Capability Set 2

ITU-T Recommendation Q.1223

(Previously CCITT Recommendation)

ITU-T Q-SERIES RECOMMENDATIONS

SWITCHING AND SIGNALLING

SIGNALLING IN THE INTERNATIONAL MANUAL SERVICE	Q.1–Q.3
INTERNATIONAL AUTOMATIC AND SEMI-AUTOMATIC WORKING	Q.4–Q.59
FUNCTIONS AND INFORMATION FLOWS FOR SERVICES IN THE ISDN	Q.60–Q.99
CLAUSES APPLICABLE TO ITU-T STANDARD SYSTEMS	Q.100–Q.119
SPECIFICATIONS OF SIGNALLING SYSTEMS No. 4 AND No. 5	Q.120–Q.249
SPECIFICATIONS OF SIGNALLING SYSTEM No. 6	Q.250–Q.309
SPECIFICATIONS OF SIGNALLING SYSTEM R1	Q.310–Q.399
SPECIFICATIONS OF SIGNALLING SYSTEM R2	Q.400–Q.499
DIGITAL EXCHANGES	Q.500–Q.599
INTERWORKING OF SIGNALLING SYSTEMS	Q.600–Q.699
SPECIFICATIONS OF SIGNALLING SYSTEM No. 7	Q.700–Q.849
DIGITAL SUBSCRIBER SIGNALLING SYSTEM No. 1	Q.850–Q.999
PUBLIC LAND MOBILE NETWORK	Q.1000–Q.1099
INTERWORKING WITH SATELLITE MOBILE SYSTEMS	Q.1100–Q.1199
Interworking with Standard-A INMARSAT system	Q.1100–Q.1109
Interworking with Standard-B INMARSAT system	Q.1110–Q.1149
Interworking with the INMARSAT aeronautical mobile-satellite system	Q.1150–Q.1199
INTELLIGENT NETWORK	**Q.1200–Q.1999**
BROADBAND ISDN	Q.2000–Q.2999

For further details, please refer to ITU-T List of Recommendations.

ITU-T RECOMMENDATION Q.1223

GLOBAL FUNCTIONAL PLANE FOR INTELLIGENT NETWORK CAPABILITY SET 2

Summary

IN Capability Set 2 (IN CS-2) is the second 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 2 (IN CS-2). This Recommendation defines the IN GFP for IN CS-2 based on the general framework for IN GFP provided in Recommendation Q.1203 [2], consistent with the scope of IN CS-2 defined in Recommendation Q.1221 [3].

This Recommendation defines:

- the IN GFP model for IN CS-2 in terms of a subset of the general IN GFP model;
- identifications and definition of the 21 IN CS-2 Service Independent Building-blocks (SIBs), including the Basic Call Process (BCP) and the Basic Call Unrelated Process (BCUP) specialized SIB;
- the use of global service logic in IN CS-2.

Companion Recommendations include the Q.120x- and Q.122x-Series of Recommendations.

Source

ITU-T Recommendation Q.1223 was prepared by ITU-T Study Group 11 (1997-2000) and was approved under the WTSC Resolution No. 1 procedure on the 12th of September 1997.

FOREWORD

ITU (International Telecommunication Union) is the United Nations Specialized Agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the 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.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

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

INTELLECTUAL PROPERTY RIGHTS

The ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. The ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, the ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

Page

Page

Page

Recommendation Q.1223

GLOBAL FUNCTIONAL PLANE FOR INTELLIGENT NETWORK CAPABILITY SET 2

(Geneva, 1997)

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 [1]. The Global Functional Plane (GFP) of the INCM is described in associated Recommendation I.329/Q.1203 [2].

IN Capability Set 2 (IN CS-2) is the second 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-2. General GFP aspects are addressed in associated Recommendations I.312/Q.1201 [1] and I.329/Q.1203 [2].

The concept of domains is introduced within this Recommendation. This concept is used to the purpose of modelling parallel processes.

2 Scope of IN global functional plane for capability set 2

The aim of the Global Functional Plane (GFP) is to satisfy the following objectives:

- to identify the set of basic network capabilities in terms of service independent building blocks (SIBs) (see clause 5);
- to develop modelling techniques to combine these SIBs to model the services/service features identified on the service plane, i.e. parallel processing techniques, granularity levels (see clause 4);
- to facilitate the mapping of the service and service features, identified on the service plane, to the set of SIBs on the global functional plane (see clause 8);
- to facilitate the mapping of SIBs onto the Distributed Functional Plane (DFP) to help identify a comprehensive set of Functional Entity Actions (FEAs) and Information Flows (IFs). SDL diagrams are provided for the individual SIB to help clarify to understand its functionality;
- only normal SIB execution is considered within the IN CS-2 GFP modelling, error recovery and error handling mechanisms are out of the scope of the IN CS-2 GFP.

3 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, 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] ITU-T Recommendation I.329/Q.1203 (1997), *Intelligent network – Global functional plane.*

[3] ITU-T Recommendation Q.1221 (1997), *Introduction to Intelligent Network Capability Set 2.*

[4] ITU-T Recommendation Q.1290 (1995), *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), *ISDN user-network interface layer 3 specification for basic call control.*

4 Global functional plane modelling for capability set 2

The INCM [1] models the GFP as shown in service implementation mapping in Figure 1. The aim of the global functional plane (GFP) is to satisfy the following two objectives:

- to identify the set of basic network capabilities in terms of Service Independent Building Blocks (SIBs);
- to describe how to combine these SIBs to model the services identified in the service plane.

A means of achieving these two objectives is to consider them as two different aspects existing at the global functional plane: the capability view and the service view.

Capability view

One aspect of the GFP identifies the individual SIBs and their discrete operations. These discrete operations contained within the SIBs can be combined to reflect the service features capabilities and facilitate the mapping of SIBs onto the Distributed Functional Plane (DFP) to help identify a comprehensive set of functional entity actions and information flows.

Service view

The other aspect of the GFP describes the methodology and tools to combine the individual *SIB operations* to implement the benchmark services and service features. This aspect of the GFP applies the SIB modelling methodology to the individual services/service features. The operations and supporting data to model these individual services/service feature are specific to that service or service feature and therefore no mapping to the underling DFP is seen to be necessary.

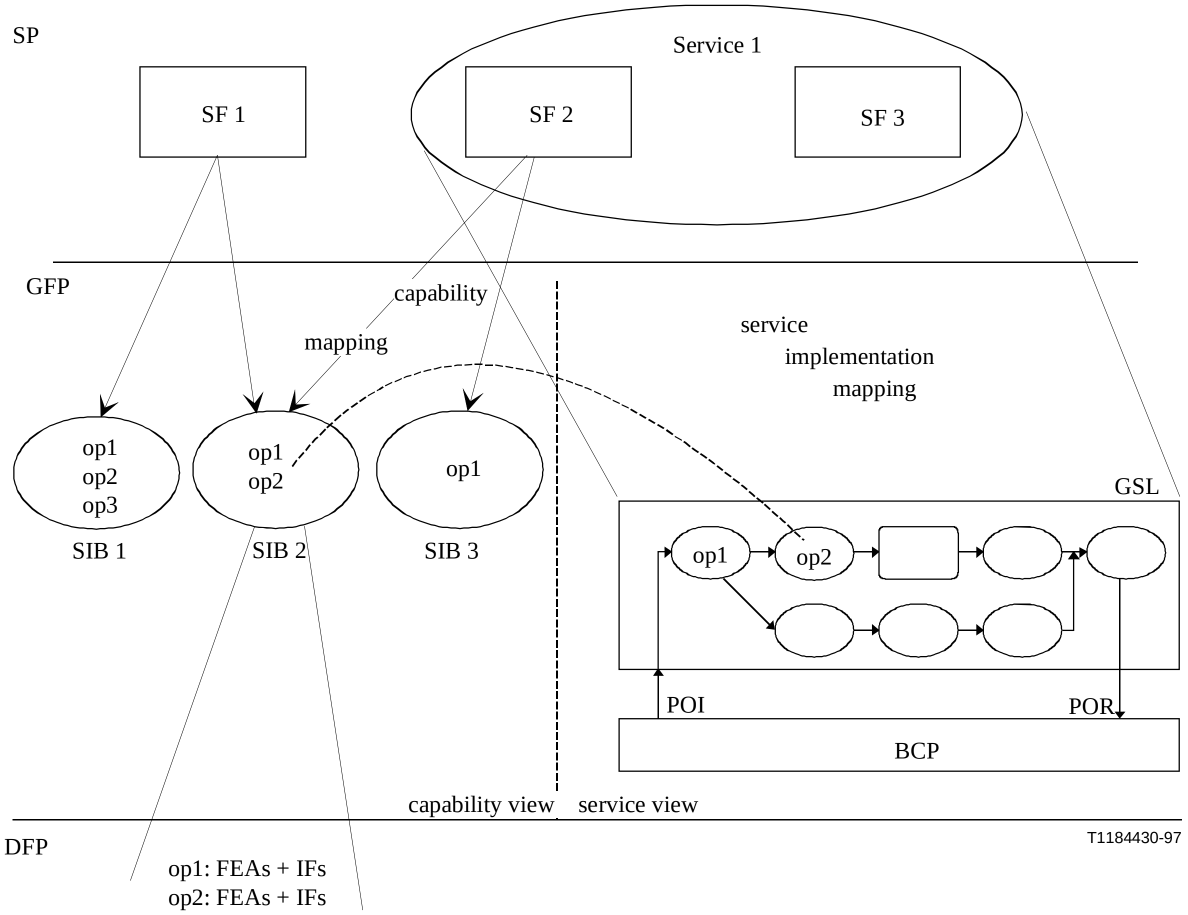


Figure 1/Q.1223 – Global functional plane model

This Recommendation specifies how the model can be used to fully define IN CS-2 service features in the GFP and aid the mapping of the GFP to the DFP.

Table 1/Q.1223 – The modelling elements within IN CS-2 GFP

Element	Description
SIB operation	A discrete, non-interruptible and atomic function performed within an SIB relating to the SIB's capability.
SIB	An SIB is a standard reusable network-wide capability consisting of SIB operations.
HLSIB	A HLSIB is a combination of SIBs containing individual operations which may subsequently be combined with other HLSIBs or SIBs to create an even higher level SIB.
BCP	A specialized SIB which provides the basic call connectivity capabilities.
Service process	A service process is considered as a combination of SIBs (containing SIB operations) or HLSIBs that represent a discrete service activity.

For the purpose of this Recommendation, with respect to service process, high level SIB and SIB operation, a drawing convention is applied as shown in Figure 2. This convention implies the following:

- A service process is symbolized by means of a rectangle.
- An HLSIB is symbolized by means of a rectangle with rounded corners.
- An SIB operation is symbolized by means of an ellipse.

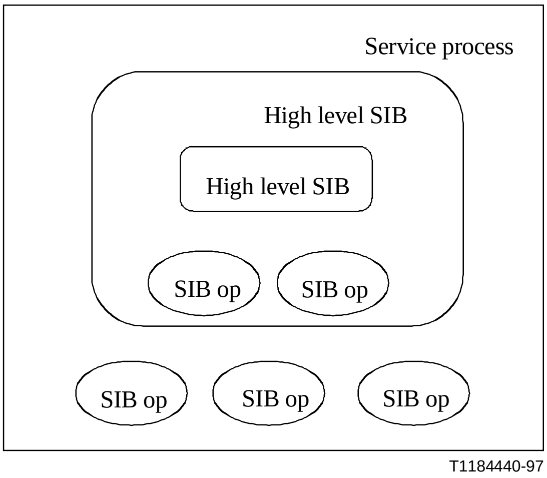


Figure 2/Q.1223 – Drawing convention used in this Recommendation

4.1 Elements on the global functional plane

4.1.1 Modelling requirements

SIB operations are considered as the lowest level of granularity. The process of mapping service features to the individual SIB operations highlights a number of prevalent combinations of SIB operations. Therefore, it is appropriate that the reusability concept of GFP have to be enhanced to accommodate these prevalent combinations of SIBs. This enhanced reusability will aid the service designer in mapping the services/service features on the GFP, enabling the service designer to save costly design time to map the services/service features to the individual SIBs on every occasion. The concept of high level SIBs is introduced to reflect this requirement.

IN CS-1 can only accommodate single service execution performed sequentially. This results in the blocking of subsequent activities until the original service execution is completed. Therefore, the ability to perform discrete service execution in parallel is necessary. The concept of parallel service processing is introduced to fulfil this requirement.

Parallel service process will enable the implementation of specific IN CS-2 services features that require parallel service processing (i.e. simultaneous announcements to different call parties, call waiting – two active threads at the same time, one monitors an incoming call).

4.1.2 Modelling elements

4.1.2.1 Granularity

An SIB is a standard reusable network-wide capability. The SIBs can be combined to realize services and service features on the service plane. The capability offered within the SIB is described by the set of SIB operations that may be invoked in the service logic.

High Level SIBs (HLSIBs) can be formed by a combination of SIBs containing individual operations. These high level SIBs may also be combined creating higher level SIBs. On the other hand, decomposition of HLSIB provide for a top-down refinement, enables the identification of the individual SIBs (see Figure 3).

A benefit to the service designer in using HLSIBs is that the number of data parameters, visible at a specific granularity level, is reduced. This is due to the fact that specific Service Support Data (SSD) and Service Instance Data (SID) are considered as of local significance to the individual SIB. Therefore, by combining these to create HLSIBs, they appear to be invisible. It is only a specific set of SSD, SID and all Call Instance Data (CID) that appears at each level of enhanced granularity due to the global nature of these data parameters.

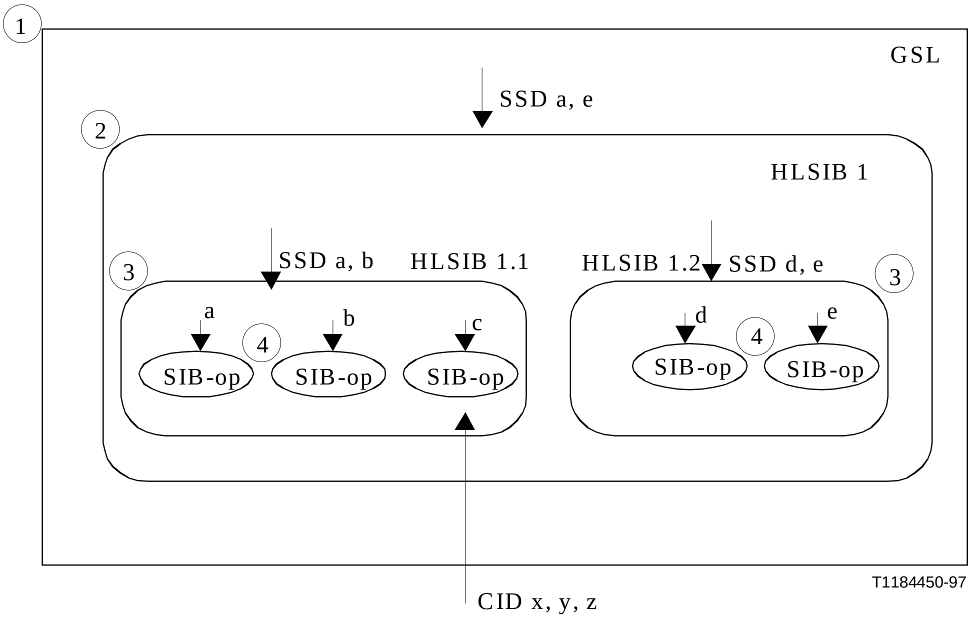


Figure 3/Q.1223 – Data abstraction using HLSIB concept

NOTE – At all levels of granularity, the three CID parameters are visible. Only specific SSD are visible at the different level of granularity.

The following data parameters are visible at the indicated granularity levels, as indicated in Table 2.

Table 2/Q.1223 – Granularity levels

Level	Number of visible parameters	Visible data parameters
4	8	a, b, c, d, e (SSD), x, y, z (CID)
3	7	a, b, d, e (SSD), x, y, z (CID)
2	5	a, e (SSD), x, y, z (CID)
1	3	x, y, z (CID)

4.1.2.2 Service process

A service process is considered as a combination of SIBs (containing SIB operations) or high level SIBs that represent a sequential service activity, e.g. authorization via a collection of PIN number and account number. A service process may initiate parallel processes that perform sequential service activities.

4.1.2.3 Parallel service processing

The concept of parallel service processing enables the initiation of subsequent service processes. A parallel service process is defined as an independent thread of service logic execution which is performed in parallel with other independent service processes.

The following requirements of parallel service processing are highlighted below:

- **Controlling service process**

  The service process of the global service logic that was invoked by a basic process (i.e. basic call process or basic call unrelated process) is called the controlling service process.

- **Initiation of other supporting service processes**

  The controlling service process can initiate other supporting service processes which are already deployed into the network. The controlling service process will have full control of its initiated service processes, referred to as supporting service processes.

- **Predefined ends**

  The supporting service process may have a stop condition. The supporting service process may have the following ends:

  1) an independent end;

  2) notify the controlling service process end;

  3) ordered to end its processing by the controlling service process.

If the service process does not have a stop condition, then it will have a lifetime beyond the controlling service process (e.g. management feature).

- **Synchronization**

  The service processes must be able to synchronize between a pair of service processes and exchange information between these service processes.

- **Awareness of service processes**

  To be able to exchange information service processes must be aware of each other existence.

- **Buffering of notifications**

  The service processes must be able to buffer synchronization notifications to process the notification at a predetermined moment.

Figure 4 illustrates the following required capabilities for parallel service processing:

1) initiate a service process;

2) send a synchronization notification, which may convey data information, from one service process to another service process;

3) receive a synchronization notification from another service process;

4) end a service process.

NOTE – This subclause on parallel service processing does not preclude the use of more generalized concepts of parallel processing. However, use of these concepts are not be used or specified in IN CS-2.

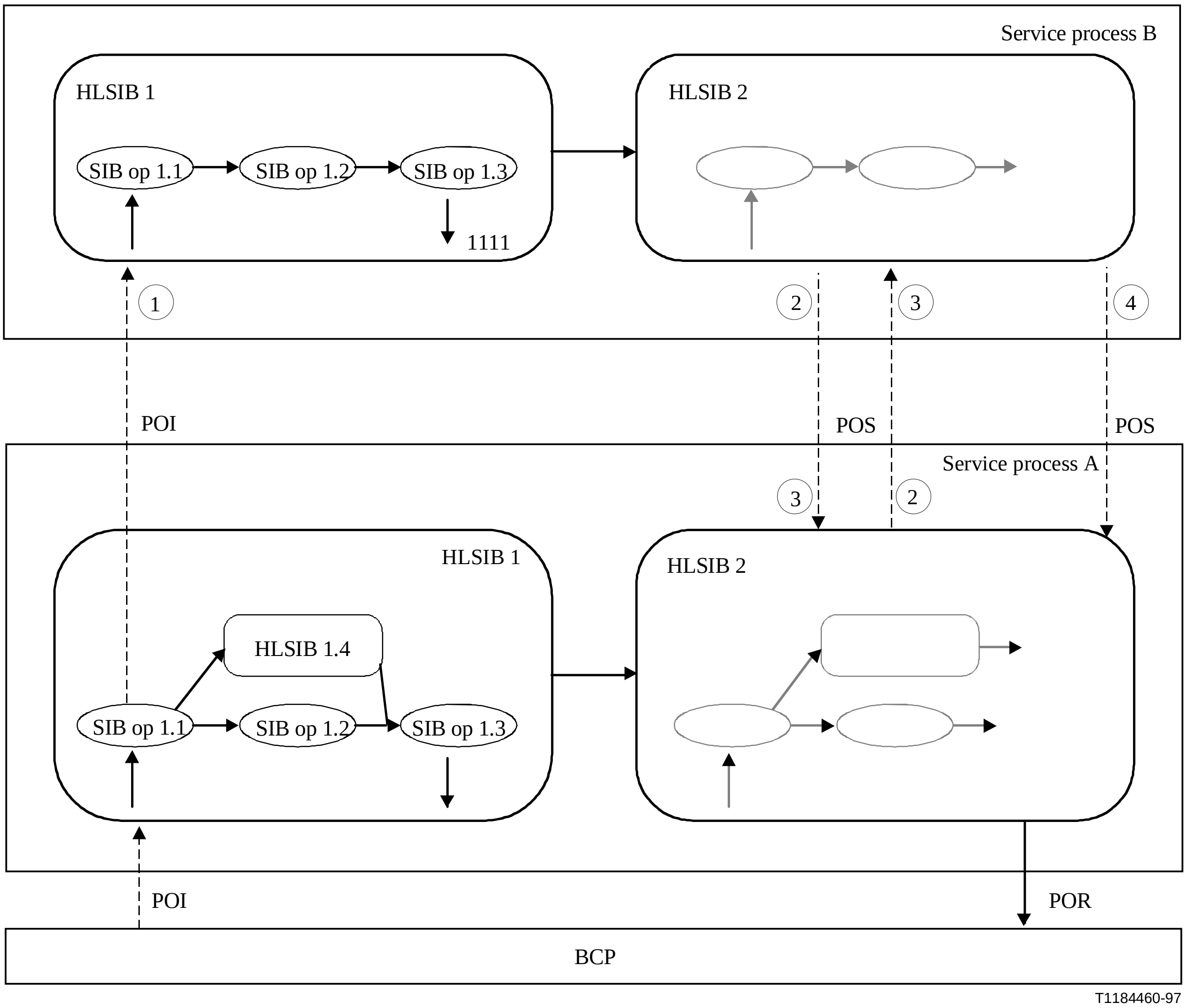


Figure 4/Q.1223 – Parallel service processing

4.2 The basic call process

The Basic Call Process (BCP) is responsible for providing basic call connectivity between parties in the network. The BCP can be viewed as a specialized SIB, which provides a set of operations, such as:

- connecting call, with appropriate disposition;
- disconnecting calls, with appropriate disposition;
- retaining CID for further processing of that call instance;
- handling multiple call parties.

The BCP is described in more detail in clause 6.

4.3 The capability view

4.3.1 Definition

The capability view describes the set of capabilities existing in one domain in terms of SIBs. The set of SIBs available in one domain define the services and SFs that can be offered to the users of that domain. Different domains may contain different set of SIBs and thus provide different capabilities or services to their users.

4.3.2 Service independent building blocks

An SIB is a standard reusable network-wide capability residing in the global functional plane used to create service features. SIBs are of a global nature and their detailed realization is not considered at this level but can be found in the Distributed Functional Plane (DFP) and the physical plane. The SIBs are reusable and can be combined to realize services and SFs of the service plane. The capability offered within the SIB is described by the set of operations that may be invoked in the SIB. The set of operations that the SIB offers constitute the SIB interface. Each operation defines a function that can be performed related to the SIB capability. Complex SIBs, e.g. those modelling persistent activities, are defined on the basis of several operations that allow the control of the activity performed by the SIB. SIBs are defined to be independent of the specific service and technology for which or on which they will be realized.

4.3.2.1 Characteristics of an SIB

SIBs have the following characteristics (see Note):

- SIBs are defined completely independent from consideration of any specific distributed functional and physical plane architectures (network implementation independent).
- Each SIB should have a unified and stable interface. The interface is described by a set of operations that describes the whole SIB activity.
- DFP interaction among FEs is not visible to the SIBs in the GFP.
- Individual SIBs must be defined using a standard methodology to allow:
  - – multi-vendor IN products to identically support them;
  - – service designers to have a common understanding of the SIB.

NOTE – No implication of importance is meant to be implied by the order of the following items:

- SIBs are the monolithic building blocks (their detailed implementation is hidden) that the service designer will use to develop new services.
- All Service Features (SFs) are described by one SIB or combination of SIBs.
- All SFs can be defined by a finite number of SIBs.
- An SIB defines one complete activity.
- SIBs are realized in the DFP by function entity actions which may reside in one or more Functional Entities (FEs).
- An SIB operation has one logical starting point and one or more logical end points. Data required by each SIB operation is defined by SIB operation support data parameters and call instance data parameters.
- SIBs are global in nature and their locations need not be considered as the whole network is regarded as a single entity in the GFP.
- SIBs are reusable. They are used without modification for other services.
- Data modelling: A formal description of SIB data is required, in order to be unambiguous, concise and precise. The use of formal data description provides for a smooth mapping to the lower planes. In order to be in line with these planes, the use of ASN.1 should be considered.
- For IN CS-2, actual network time and date are assumed to be available to all SIBs in the GFP and it does not have to be passed through call instance data.

4.3.2.2 Data parameters for SIBs

By definition, SIBs are independent of the service/SF they are used to represent. They have no knowledge about other 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 an 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.

Data parameters consist of input and output parameters. Two general types of data parameters are required for each SIB operation, dynamic parameters called Call Instance Data (CID) and static parameters called Service Support Data (SSD) and Service Instance Data template (SID).

Differentiation of formal SIB parameters from actual parameters introduces more flexibility assigning the SIB parameter data type. The formal SIB parameters are the parameters that will be used for SIB descriptions in this Recommendation. Actual SIB parameters only occur in SIB instances in specific Global Service Logic (GSL).

4.3.2.2.1 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 (e.g. calling line identification);
- generated by an SIB operation (e.g. a translated number);
- or entered by the subscriber (e.g. a dialled number or a PIN code).

4.3.2.2.2 Service Support Data (SSD)

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

4.3.2.2.3 Service Instance Data template (SID)

Service instance data template defines data related to a service subscriber's profile that exists before the service is invoked and can be modified and updated as a result of the service processing activity. This type of data can be read within the service execution and be stored to be used in further service invocations.

4.3.2.2.4 Assignments of formal and actual SIB parameters

The formal parameters of an SIB operation are assigned during the SIB definition phase. This description will be included in this Recommendation. The actual SIB parameters are assigned during the Service design phase, where the GSL of a service is designed.

4.3.3 Methods to describe SIBs

The SIBs provide the modularity within the global functional plane that is required by the definition and objectives of the IN concept. In order to effectively progress such studies, a method is required to characterize and technically describe the SIBs.

Techniques analogous to those used in the 3-stage service definition methodology [5], i.e. prose description, static description, and dynamic description, are appropriate.

The procedure outlined in Figure 5 can be used to determine if new SIBs are required to support new services.

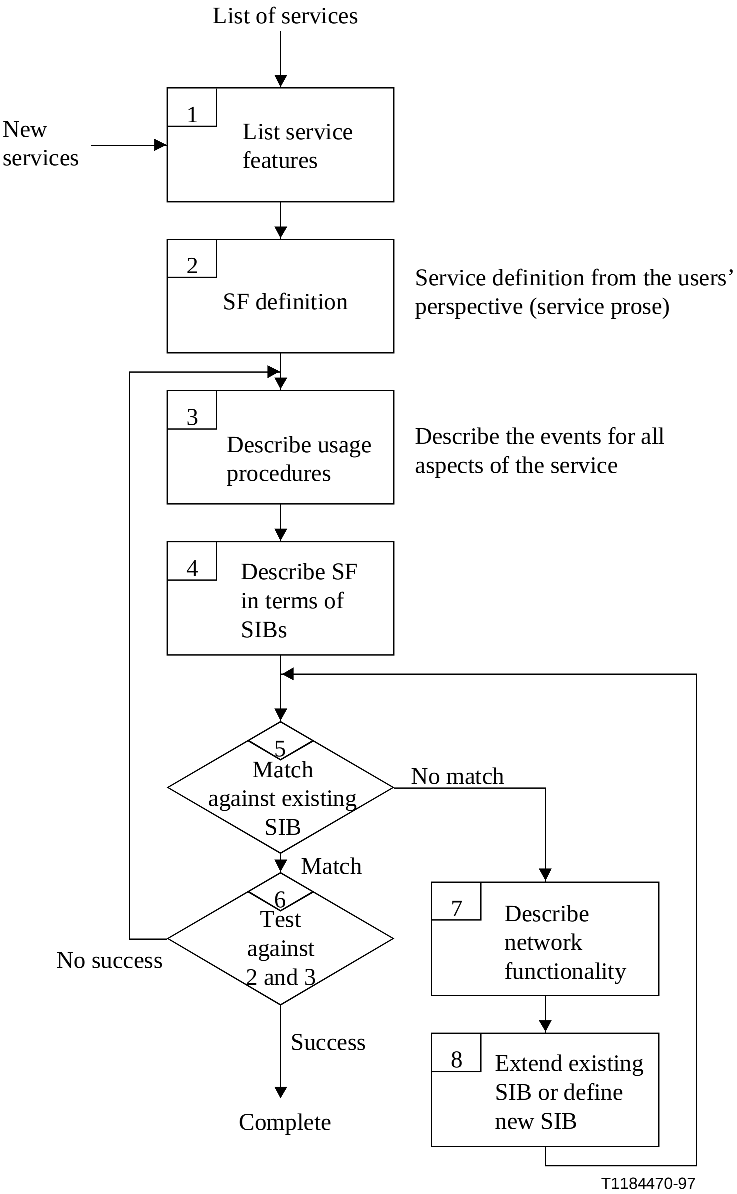

List of services
New services
1
List service features
2
SF definition
Service definition from the users' perspective (service prose)
3
Describe usage procedures
Describe the events for all aspects of the service
4
Describe SF in terms of SIBs
5
Match against existing SIB
No match
Match
6
Test against 2 and 3
No success
Success
Complete
7
Describe network functionality
8
Extend existing SIB or define new SIB
T1184470-97

Figure 5/Q.1223 – Flow chart to identify service independent building blocks

The following terms are used in the SIBs' identification method.

4.3.3.1 Definition

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

4.3.3.2 Potential service applications

Service examples of where this SIB can be used.

4.3.3.3 Interface

Identifies the SIB operations that constitute the SIB interface, and provides a textual description on the utilization of the operations, and how they relate to the whole activity of the SIB.

The SIB interface is defined by four elements: operations, input, output and graphic representation.

4.3.3.3.1 Operation

Description of actions performed by the SIB operation.

4.3.3.3.2 Input

Input to each SIB is specified as three distinct elements:

- One logical starting point;
- Parameters:
  - service support data which defines parameters which are specified by the service description,
  - service instance data template which are specific to a service subscriber's profile,
  - call instance data which are specific to that call instance;
- Inter-process communication which defines the data that may be received.

Output

Output from each SIB is specified as two distinct elements:

- One or more logical end points;
- Parameters:
  - service data instance template which defines data parameters specific to a service subscriber's profile which results from the execution of that SIB operation and are required by other SIB operations to complete the call service instance,
  - call instance data which defines data parameters specific to that call instance which results from the execution of that SIB operation and are required by other SIB operations to complete the call service instance;
- Inter-process communication which defines the data that may be sent.

Graphic representation

A graphic representation is used to describe each operation contained within an SIB interface. It is illustrated in Figure 6. Each SIB operation is characterized by having input and output parameters, one input logical flow and one or more output logical flows. 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. Input and output parameters are identified by the dashed arrows at the top of the diagram and are specified beside the dashed arrow. For both, input and output parameters, SSD, SID and CID type is declared beside the respective parameters. Similarly, POS are specified below the diagram.

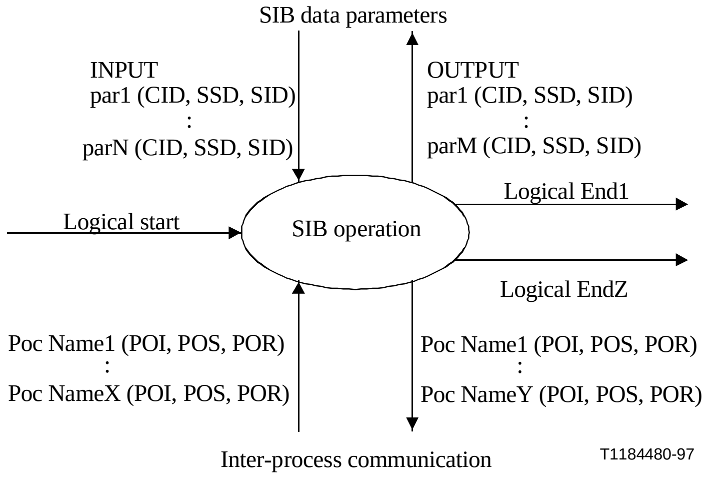


Figure 6/Q.1223 – SIB operation graphical representation

4.3.3.4 Graphic representation of the SIB

A graphic representation is used to describe the operations that the SIB interface contains. It is illustrated in Figure 7. Each SIB is defined by the operations it offers through its interface.

SIB

– Operation 1

:

– Operation Y

Figure 7/Q.1223 – SIB graphic representation

4.3.3.5 SDL diagram

This diagram gives a graphic representation of the stage 1 description of the SIB using SDL macro diagrams [6]. The SDL diagram is given for explanatory reasons in the stage 1 description if required to specify the functionality of the SIB.

NOTE – Within IN capability set 2, no SDL diagrams are used to illustrate the functionality of the SIBs. Instead of SDL diagrams in the SIB stage 1 description references to the SDL diagrams are included in the stage 1 descriptions if required.

4.4 Interaction management

4.4.1 Interaction between SIBs

Interaction management in the GFP consists in the investigation whether two or more SIBs have impact on each other. It is performed by comparing the descriptions of different SIBs, using the following rules:

- Each SIB works on a set of basic properties, such as resources, data, etc.; there is only a very low probability for an SIB based on the manipulation of resources to interact with an SIB based on the manipulation of data.

- The mutual use of the same data or resources by two SIBs should be exactly identified, as it may cause problems, for instance when the order in which the SIBs could be active is not defined.
- When a new SIB is being defined, the already existing SIBs have to be taken into account to avoid two SIBs performing the same, or almost the same, tasks.

4.4.2 Interaction handling methods in the SIB definition phase

Interaction may occur when two SIBs use the same data or resources, which may occur either when both SIBs are active at the same time or when they are active sequentially.

For instance, one may consider the two SIBs "CHARGING" and "USER INTERACTION". The first SIB increments a charging counter every minute, for a given call. The second SIB has to present the charging counter value to the user at the end of the call. If no specific precaution is taken, "USER INTERACTION" SIB may read the counter value before the final update by the "CHARGING" SIB. This specific precaution is a statement that could be written: "USER INTERACTION" shall not be active on a charging data before or at the same time that "CHARGING" is active.

Interaction may be solved by preventing such a kind of situations, which leads to the following rules:

- Two SIBs should be prevented from manipulating the same data or resources (the mutual use of the same data may cause problems when the sequence in which SIBs should be activated is not defined).
- When two SIBs are felt in an interaction situation, this situation should be prohibited from decreasing one of the SIBs' functionality, that leads to increase its atomicity and to define smaller SIBs with more specific tasks.

4.5 The service view

4.5.1 Definition

The service view describes how IN services and service features can be described in GFP, without regarding distribution and real time aspects of network capabilities. These service features are built out of communicating service processes, in which each service process can use operations offered by SIBs, in an order that is described by its GSL.

4.5.2 Global service logic

4.5.2.1 General

The GSL can be defined as the "glue" that defines the order in which SIB operations will be chained together to build service processes to accomplish service features. Each instance of global service logic is (potentially) unique to each individual call, but uses common elements, comprising specifically:

- interaction points (POI, POS and POR) of service processes, including the BCP service process;
- SIB operations;
- logical connections between SIB operations, and between SIB operations and service process interaction points;
- input and output data parameters, service support data and call instance data defined for each SIB.

Based upon the functionality of these common elements, global service logic will "chain together" these elements to provide a specific service.

In order to more completely demonstrate how GSL operates, a generic example of a service is illustrated in Figure 8. This diagram shows that specific SIB chains launched from a designated POI, are activated in a particular order and are returned to the appropriate PORs, as required by the GSL. To avoid complexity, the SIB data parameters are not shown.

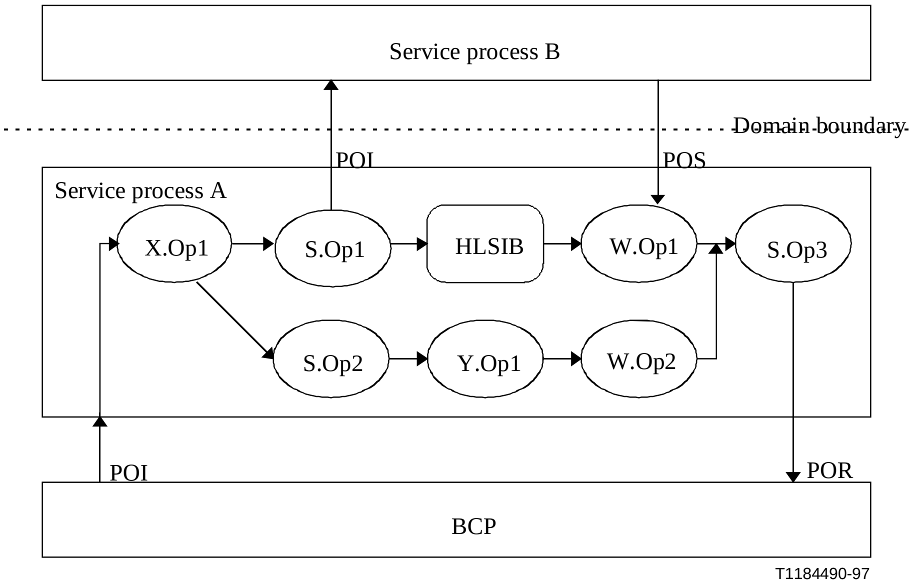


Figure 8/Q.1223 – Modelling service features by using SIB operations and HLSIBs

4.5.2.2 Relationship between GSL and BCP

Global service logic on the global functional plane views basic call process as a single resource, which offers a set of operations. 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) synchronization – which is represented by POSs;

iii) 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 (a specialized POS);

ii) synchronization – which is represented by POSs;

iii) data – which is represented by call instance data that have been defined by one or more SIBs on an 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.

4.5.2.3 Relationship between global service logic and SIBs

The remaining components of GSL needed to define a service/SF are the collection of SIB operations (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.

4.5.3 SIB operations

Service Independent Building Blocks (SIBs) offer a set of operations to be used to describe service features in the service view. An example is shown in Figure 8. In this example, the BCP invokes service process A, which first performs operation 1 offered by SIB X. Depending on the result, one of both outlets is chosen. If the top outlet is chosen, service process A spawns a new service process B, executes an HLSIB and is waiting for a POS from service process B which is reflected by the operation 1 offered by SIB W. When the other outlet is chosen, this service process executes operation 2 of SIB S, operation 1 offered by SIB Y, operation 2 offered by SIB W and service process A will return a POR to the BCP which is reflected by operation 3 offered by SIB S.

4.5.4 Characteristics of an HLSIB

High level SIBs (HLSIBs) are, as normal SIBs, a reusable part of a service feature, but composed out of SIB operations and other HLSIBs which can be executed sequentially. An HLSIB stands for abstraction by hiding service logic and parts of the SSD that is considered to be local to the HLSIB. HLSIBs have the following additional characteristics:

- HLSIBs can be composed out of other HLSIBs and SIB operations only.
- A certain HLSIB can not be used as a component within the same HLSIB, i.e. no recursive use is possible.
- The lowest level of HLSIBs contains SIB operations only, i.e. no further detail is visible on the GFP.
- One of the (HL)SIBs within an HLSIB is the first to be executed; therefore, HLSIBs have only one entry point (logical start), the same as with normal SIBs. But, as with normal SIBs as well, HLSIBs can have one or more exit points (logical ends).
- An HLSIB resides always in a single domain. An HLSIB can not span multiple domains, because high level SIBs are within a single service process.

4.5.5 Characteristics of a service process

A service process represents a chain of SIBs or HLSIBs which are executed sequentially, but in parallel with other service processes. Service processes are spawned via a POI and synchronized via POSs. Data can be communicated between service processes via CID. A service process is always contained within one domain. But, a process in one domain is able to spawn new processes and to communicate with processes in other domains or its own domain. The POCs and CID crossing the border of two domains is part of their logical interface. The BCP can be regarded as a specialized service process.

Service processes have the following characteristics:

- Service processes can only be composed out of HLSIBs and SIBs that are executed sequentially.
- One of the (HL)SIBs within a service process is the first to be executed; therefore, service processes have only one entry point initiated via a POI.
- Communication between service processes (including toward the BCP) can be performed by POSs.
- Service processes need mechanisms to send, receive and process POCs and to use the attached data.

- Call Instance Data (CID) is considered to be local to a service process, but global within that service process. Data exchange is performed explicitly via POCs.
- A service process resides always in a single domain. A service process can not span multiple domains.

4.5.6 Communication between service processes

New mechanisms have to be provided to support communication requirements as the result of the concept of decomposition of the GSL in several service processes. The basic mechanisms required are the creation of service processes, and the interchange of data between different service processes:

- **Creation of service processes**: To create new service processes, a specific mechanism has to be provided.
- **Interchange of data between service processes**: Service processes may need to interact to synchronize and/or interchange data. A mechanism to support this requirement has to be provided.

Communication between two service processes can be achieved via Points of Synchronization (POSs). A POS is a functional interface between service logic of two service processes over which asynchronous communication is initiated. This means that a particular SIB in the sending service process has capabilities to send a synchronize signal to a different service process that is executed in parallel. After the synchronize signal has been send this service process can continue its execution. The receiving service process, however, has to wait until the synchronize signal has arrived. This means that the execution of a particular SIB in the receiving service process has to be suspended until the synchronize signal has been received. If the receiving service process is not yet suspended at the time the synchronize signal has been received, the service process must buffer the POS. If exception handling is initiated during the service logic execution, then all messages to the particular service process will be discarded.

Full synchronous communication can be achieved by both SIBs of the two service processes that are performing a handshake with two POSs.

Two specialized SIBs having send and receive capabilities to send and receive POSs. Via these two SIBs, asynchronous communication can take place between two parallel executing service processes. Full synchronous communication can be achieved by using both SIBs in both service processes that are performing a handshake.

The POI can also be regarded as a form of synchronization. This means that a particular SIB in the initiating service process has spawn capabilities to initiate via a POI a new service process to be executed in parallel.

4.5.7 Domains

Domains, which encapsulate predefined roles in the IN structured network, are modelled by service processes that communicate over domain boundaries. An example is shown in Figure 8, in which service process "A" of domain "A" communicates with service process "B" of domain "B", by means of POCs: POIs and POSs. In this example, SIB operation "S.Op1" spawns a new service process in domain "B" via a POI, and both service processes synchronize by sending and receiving a POS between SIB operation "W.Op1" and another SIB operation within service process B. These POCs are also used to explicitly exchange data between service processes and therefore between domains. So, the set of POIs and POSs between two domains define exactly their logical interface. Furthermore, to have full control within a certain domain, service processes may not exceed domain boundaries.

4.6 Terminology

The following terms used in this Recommendation are more fully defined in Recommendations I.329/Q.1203 [2] and Q.1290 [4]:

BCP	Basic Call Process
BCUP	Basic Call Unrelated Process
CID	Call Instance Data
CLI	Calling Line Identification
POI	Point of Initiation
POR	Point of Return
POS	Point of Synchronization
SID	Service Instance Data template
SSD	Service Support Data

5 IN CS-2 Service Independent Building Blocks (SIBs)

The following list of SIBs have been identified as required to support the list of targeted IN CS-2 services and service features identified in Recommendation Q.1221 [3]:

- ALGORITHM;
- AUTHENTICATE;
- CHARGE;
- COMPARE;
- DISTRIBUTION;
- END;
- INITIATE SERVICE PROCESS;
- JOIN;
- LOG CALL INFORMATION;
- MESSAGE HANDLER;
- QUEUE;
- SCREEN;
- SERVICE DATA MANAGEMENT;
- SERVICE FILTER;
- SPLIT;
- STATUS NOTIFICATION;
- TRANSLATE;
- USER INTERACTION;
- VERIFY;
- BASIC CALL PROCESS (BCP);
- BASIC CALL UNRELATED PROCESS (BCUP).

The stage 1 SIB descriptions that follow reflect the understanding of the logical function of each SIB in its role of supporting IN CS-2 services and service features. The BCP and BCUP, which are viewed as specialized SIBs, are described in clause 6.

An explanation of the format of the stage 1 SIB descriptions can be found in 4.3.3.

5.1 ALGORITHM

5.1.1 Definition

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

5.1.2 Potential service applications

The following services are the potential services that use this capability:

- mass calling;
- televoting.

5.1.3 Interface

This SIB takes specified input 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. It consists of one single SIB operation:

- apply algorithm.

5.1.3.1 Apply algorithm

This SIB operation takes specified input data and applies the specified mathematical algorithm to it to produce the corresponding data result.

5.1.3.1.1 Input

Logical start

Indicates the logical start of execution for the SIB.

Input data parameters

- Type

  Specifies the type of algorithm for this SIB operation:

  1) increment;

  2) decrement.

  This parameter behaves as service support data.

- Value

  Specifies the increment or decrement value to be used when applying the SIB operation (e.g. 1, 2, etc.). This parameter behaves as service support data.

Inter-process communication

None.

5.1.3.1.2 Output

Logical end

- Outlet success;
- Outlet error.

Output data parameters

- Data

  Contains the resultant value after the SIB has completed. This parameter behaves as 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 algorithm:

  1) invalid type;

  2) invalid value.

Inter-process communication

None.

5.1.3.1.3 Graphical representation

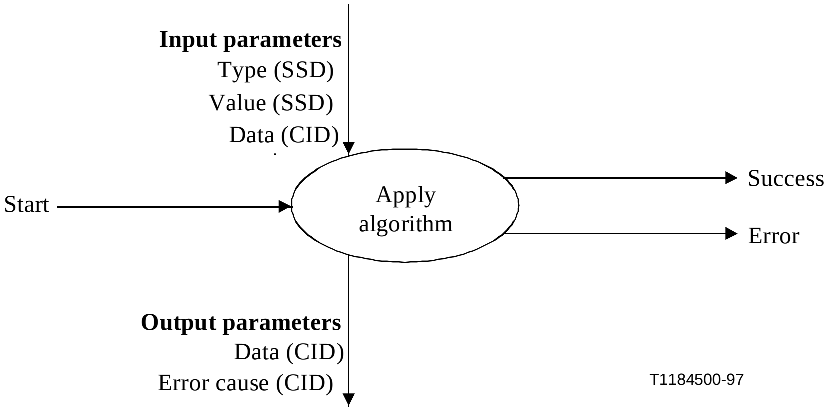


Figure 9/Q.1223 – Apply algorithm SIB operation

5.1.4 SIB graphical representation

ALGORITHM SIB

– Apply algorithm

Figure 10/Q.1223 – ALGORITHM SIB

5.2 AUTHENTICATE

5.2.1 Definition

This SIB provides the functionality necessary to establish a relationship between service logic and service data based on a specific user identity. This identity is used by subsequent service data access operations to determine if the user identity has the necessary access privileges to perform the requested operations.

5.2.2 Potential application services

The following services are the potential services that use this capability:

- All IN CS-1 services which access service data will require AUTHENTICATE.

5.2.3 Interface

The AUTHENTICATE SIB consists of two operations: authenticate data user and end authenticated relationship. Authenticate data user provides all the processing needed to establish an authorized relationship for a given user identity between service logic and service data. Authenticate end terminates an established authorized relationship.

The SIB operation authenticate data user must be performed before any subsequent service data access operations for a given user identity.

The stage 2 SDL representation of this SIB may be helpful in understanding its operation. Refer to 11.2.2.3/Q.1224.

5.2.3.1 Authenticate data user

This SIB operation establishes an authorized relationship between service logic and service data. Once the relationship has been established, the service logic may perform service data operation against the service data. The user identity which has been established determines the degree of access to the service data during the subsequent access operations.

5.2.3.1.1 Input

Logical start

Indicates the logical start of the SIB operation.

Input data parameters

- Authenticate name

  Specifies which user identity is to be authenticated.

  This parameter is mandatory if the "authentication mechanism Id" parameter has the value "simple" or "strong" and is optional for value "external". It is not used if the value is "public".

  This parameter behaves as service support data or as call instance data.

- Authenticate password

  Specifies the password to be used to authenticate the user identity specified in "authenticate name".

  This parameter is only used if the "authentication mechanism Id" parameter has the value "simple".

  This parameter behaves as service support data or as call instance data.

- Authentication mechanism Id

  Specifies the mechanism to be used to establish the authorized relationship. It may take one of the following values:

  1) public:

     This value implies that no specific authenticate name or authenticate password is specified by the client. The mechanism defaults to a "public access" relationship for a user between the service logic and the service data. It is equivalent to service data providing the authenticate name ("public") and an appropriate authenticate password.

2) simple:

This value implies that the user identity specified in "authenticate name" is authenticated using the password value in "authenticate password". If no password is supplied, the value defaults to a "public access" relationship for that user between the service logic and the service data.

3) strong:

This value implies that the user identity specified in "authenticate name" is authenticated using a bind token which is generated using some specified algorithm and (optionally) a certification mechanism.

4) external:

This value implies that the authentication uses an externally supplied procedure and parameters.

This parameter behaves as service support data.

Inter-process communication

None.

5.2.3.1.2 Output

Logical end

- Outlet success

  This outlet indicates that the data user has been authenticated to access service data.

- Outlet error

  This outlet indicates that the authenticate data user operation caused an error. The reason of this error is indicated by the output parameter "error cause".

Output data parameters

- Error cause

  Identifies the specific condition that caused an error during the operation of the SIB. The following errors have been identified for the authenticate data user SIB operation:

  1) busy;

  2) inappropriate authentication;

  3) invalid credentials.

  This parameter behaves as call instance data.

- Authorized relationship Id

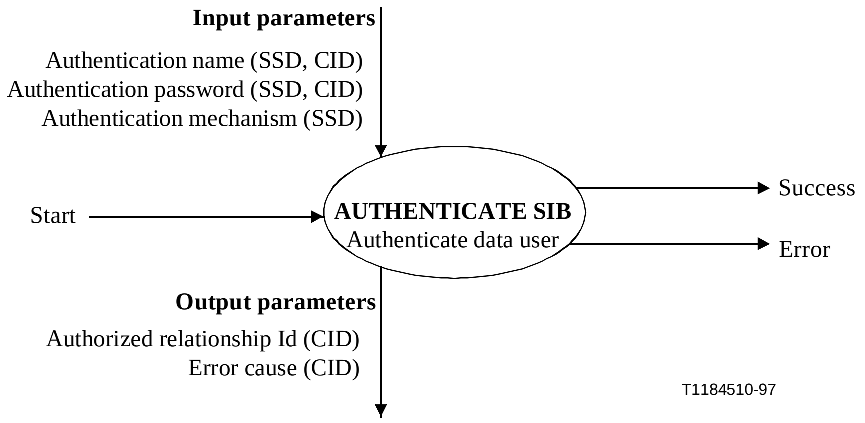
  Specifies the established relationship against which data access operations can be applied without further authentication.

  This parameter behaves as call instance data.

Inter-process communication

None.

5.2.3.1.3 Graphical representation



Input parameters
Authentication name (SSD, CID)
Authentication password (SSD, CID)
Authentication mechanism (SSD)
Start
AUTHENTICATE SIB
Authenticate data user
Success
Error
Output parameters
Authorized relationship Id (CID)
Error cause (CID)
T1184510-97

Figure 11/Q.1223 – Authenticate data user SIB operation

5.2.3.2 End authenticated relationship

This SIB operation ends the established authenticated relationship.

5.2.3.2.1 Input

Logical start

- Indicates the logical start of the SIB operation.

Input data parameters

- Authorized relationship Id

  Specifies the established relationship against which data access operations were applied.

  This parameter behaves as call instance data.

Inter-process communication

None.

5.2.3.2.2 Output

Logical end

- Outlet success

  This outlet indicates that the authenticated relationship has been successfully ended.

- Outlet error

  This outlet indicates that the End authenticated relationship operation caused an error. The reason of this error is indicated by the output parameter "error cause".

Output data parameters

- Error cause

  Identifies the specific condition that caused an error during the operation of the SIB. The following errors have been identified for the end authenticated relationship SIB operation: invalid authorized relationship Id.

Inter-process communication

None.

5.2.3.2.3 Graphical representation

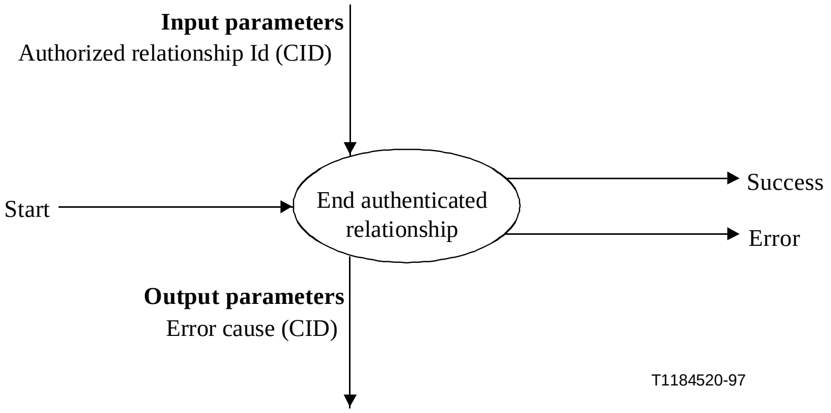


Figure 12/Q.1223 – End authenticated relationship SIB operation

5.2.4 SIB graphical representation

AUTHENTICATE SIB

– Authenticate data user

– End authenticated relationship

Figure 13/Q.1223 – AUTHENTICATE SIB

5.3 CHARGE

5.3.1 Definition

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

In the IN CS-2 context, this means that the SIB provides the functionality for producing the data to be recorded physically.

In general, this involves identifying:

- the call for which usage is to be observed;
- the resources used, plus related data, e.g. begin and end of usage;
- operational data, e.g. restrictions relative to the resource.

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

5.3.2 Potential application services

– Any service which requires specific IN charging.

5.3.3 Interface

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 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.

Charge initiation is always required and must precede the provision of charge reports where required. Therefore, the SIB operation "Charge reporting" will always be preceded by the SIB operation "Charge initiation", but "Charge initiation" operation may be followed by "Charge reporting" operation.

Charge initiation ⇨ Charge reporting

5.3.3.1 Charge initiation

This SIB operation is used to initiate IN charging on any call requiring it.

5.3.3.1.1 Input

Logical start

Indicates the logical start of the SIB operation.

Input data parameters

- Reference

  Specifies the reference to its succeeding charge reporting SIB operation.

  This parameter behaves as service support data.

- Number of accounts to charge

  This parameter behaves as service support data.

- Account

  Each account is specified by two parameters, as follows:

  1) Number
     Specifies one of the following:

     - Line(s)

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

       This parameter behaves as call instance data.

     - 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.

       This parameter behaves as call instance data.

     - Fixed account

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

  2) Per cent (%)

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

     This parameter behaves as service support data.

- Resource type

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

  This parameter behaves as service support data.

- Units

  Specifies a premium value for the specified resource type.

  This parameter behaves as service support data.

- Service/service feature identifier

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

  This parameter behaves as service support data.

- Type

  When specified, indicates the type of charging that pulse metering is associated with the called or calling line.

  This parameter behaves as call instance data.

Inter-process communication

None.

5.3.3.1.2 Output

Logical end

- Outlet success

  This outlet indicates that the charging was initiated successfully.

- Outlet error

  This outlet indicates that the charge initiation operation caused an error. The reason of this error is indicated by the output parameter "error cause".

Output data parameters

- Error cause

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

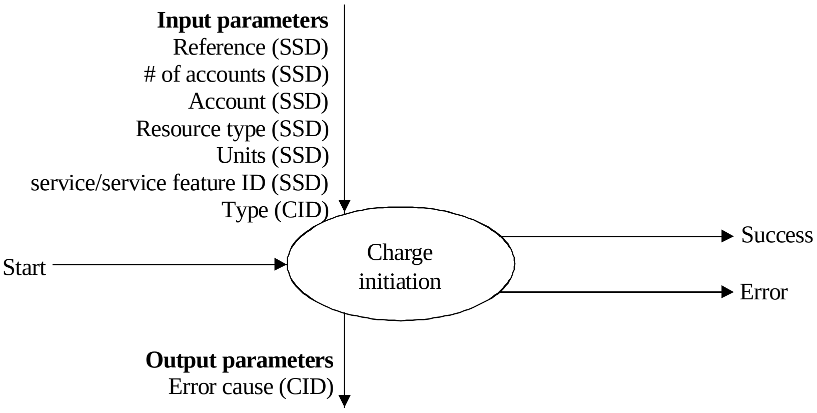
  1) invalid account to charge;
  2) invalid percentage allocation (>100, <0);
  3) invalid sum of percentage allocations (not equal to 100);
  4) invalid service/service feature;
  5) invalid resource;
  6) invalid units;
  7) system fault – unable to write record.

  This parameter behaves as call instance data.

Inter-process communication

None.

5.3.3.1.3 Graphical representation



T1184530-97

Figure 14/Q.1223 – Charge initiation SIB operation

5.3.3.2 Charge reporting

This SIB operation provides the service logic with a charge report. Upon receipt of this information, the outlet "end" will be selected.

5.3.3.2.1 Input

Logical start

Indicates the logical start of the SIB operation.

Input data parameters

- Reference

  Specifies the reference to its preceding charge initiation SIB operation.

  This parameter behaves as service support data.

- Type

  When specified, indicates the type of charging that pulse metering is associated with the called or calling line.

  This parameter behaves as call instance data.

Inter-process communication

None.

5.3.3.2.2 Output

Logical end

- Outlet end

  This outlet indicates that the charge report has been received and charging has ended.

- Outlet error

  This outlet indicates that an error was encountered.

Output data parameters

- Charge report

  Specifies the charge report. Each report is related to a specific charged party.

  This parameter behaves as call instance data.

- Error cause

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

  1) invalid account to charge;
  2) invalid percentage allocation (> 100, < 0);
  3) invalid sum of percentage allocations (not equal to 100);
  4) invalid service/service feature;
  5) invalid resource;
  6) invalid units;
  7) system fault – unable to write record.

  This parameter behaves as call instance data.

Inter-process communication

None.

5.3.3.2.3 Graphical representation

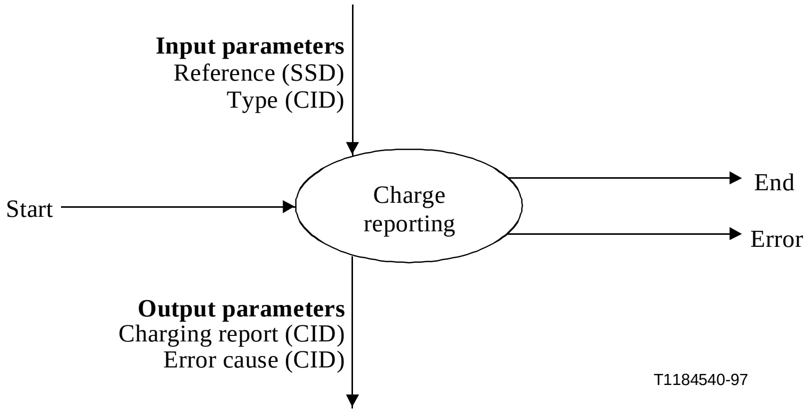


Figure 15/Q.1223 – Charge reporting SIB operation

5.3.4 SIB graphical representation

CHARGE SIB

– Charge initiation

– Charge reporting

Figure 16/Q.1223 – CHARGE SIB

5.4 COMPARE

5.4.1 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.

The identifier and the reference value should have the same comparison type. Otherwise, the logical end of the SIB is error.

5.4.2 Potential service applications

The following services are the potential services that use this capability:

- time dependent routing;
- CCBS;
- credit card calling.

5.4.3 Interface

This SIB compares an identifier to a specified reference value. The compare SIB interface consists of one single SIB operation:

- Compare information.

5.4.3.1 Compare information

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

It can be used for:

- comparing an identifier to a specified reference value or to compare collected digits against a predefined service code also registered as CID: For instance, for checking that the current number of calls is less than the maximum number authorized.
- 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.

5.4.3.1.1 Input

Logical start

Indicates the logical start of execution for the SIB operation.

Input data parameters

- Comparison type

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

  1) Identifier value – Compare the identifier against the reference value.
  2) Time – Compare network time to the reference value. Network time is specified as:
     - time of day;
     - day of week; or,
     - day of year.

  This parameter behaves as service support data.

- Data

  This pointer specifies the data to be compared to the reference value. This parameter behaves as call instance data.

- Reference value

  Specifies the value against which the comparison will be made. This parameter behaves as service support data or as call instance data.

Inter-process communication

None.

5.4.3.1.2 Output

Logical end

- GREATER THAN the value;
- LESS THAN the value;
- EQUAL TO the value;
- Error.

Output data parameters

- Error cause

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

  1) invalid identifier;

  2) invalid reference value.

  This parameter behaves as call instance data.

Inter-process communication

None.

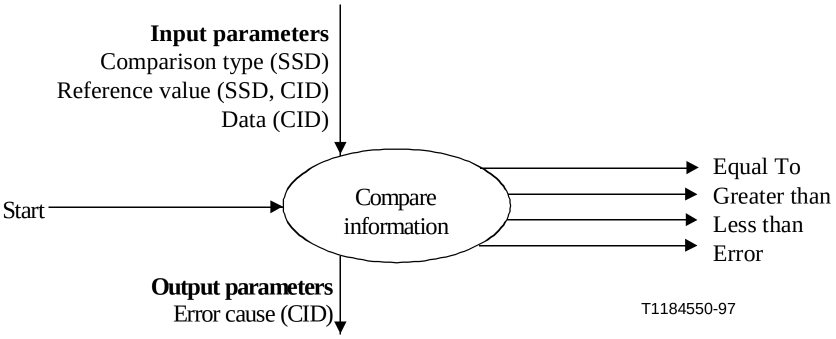


Figure 17/Q.1223 – Compare information SIB operation

5.4.4 SIB graphical representation

COMPARE SIB

– Compare information

Figure 18/Q.1223 – COMPARE SIB

5.5 DISTRIBUTION

5.5.1 Definition

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

5.5.2 Potential service applications

The following services are the potential services that use this capability:

- mass calling;
- televoting;
- freephone.

5.5.3 Interfaces

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. The SIB exists of one single SIB operation:

- Apply distribution.

5.5.3.1 Apply distribution

5.5.3.1.1 Input

5.5.3.2 Logical start

Indicates the logical start of execution for the SIB operation.

Input data parameters

- Algorithm type

  1) percentage;
  2) sequential;
  3) time of day;
  4) day of week.

  This parameter behaves as service support data.

  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

  This parameter behaves as service support data.

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

- Algorithm parameters

  1) If type = percentage

     For each logical end

     – percentage (sum over all paths must equal 100).

  2) If type = sequential

     – None

  3) 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.

4) 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.

This parameter behaves service support data.

Inter-process communication

None.

5.5.3.2.1 Output

Logical end

- 1
- 2
- ...
- N [where N = number of logical ends (Item 2 of input parameter)]
- Error

Output data parameter

- Selected logical end

  This parameter behaves as 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:

  1) invalid type;
  2) invalid percentage allocation (> 100, < 0);
  3) invalid sum of percentage allocations (not equal to 100);
  4) invalid number of logical ends;
  5) missing time period;
  6) overlap of the time across logical ends;
  7) missing days;
  8) overlap of days across logical ends.

  This parameter behaves as call instance data.

5.5.3.2.2 Graphical representation

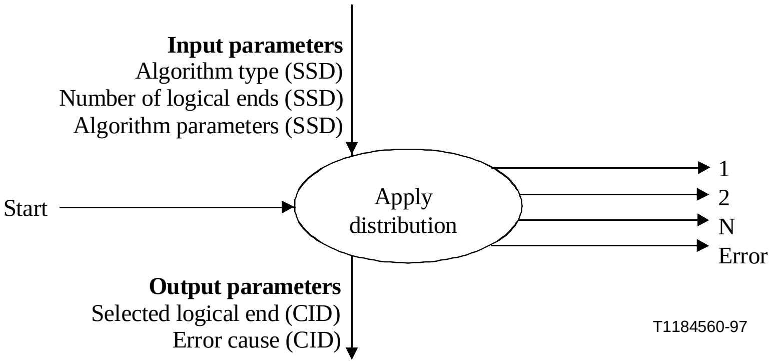


Figure 19/Q.1223 – Apply distribution SIB operation

5.5.4 SIB graphical representation

DISTRIBUTION SIB

– Apply distribution

Figure 20/Q.1223 – DISTRIBUTION SIB

5.6 END

5.6.1 Definition

Indicates the normal end of an executing service process, or part of a service process in case of multiple threads.

5.6.2 Potential application services

The following services are the potential services that use this capability:

- UPT;
- VPN;
- customized call routing (public and private).

5.6.3 Interface

The END SIB interface consists of the following SIB operation:

- End, which resumes a service process if IPC monitors are set, else it will end the parallel service process execution.

5.6.3.1 End

This SIB operation indicates the normal end of an executing parallel service process. But, in case of synchronous multiple threads, this SIB operation marks the end of such a thread, after which the service process is suspended for new POSs to be processed.

The END SIB operation is intended to be used in both controlling and supporting mode. The end SIB operation in the controlling mode indicates that the supporting service logic is to be terminated and reports this fact to the supporting service logic. The end SIB operation in the supporting mode will receive the termination request from the controlling service logic. The supporting side will terminate the service execution.

5.6.3.1.1 Input

Logical start

Indicates the logical start of the SIB operation.

Input data parameters

- Clear POSs

  Specifies a set of POSs to be removed from the buffer if they are present. This parameter behaves as service support data or as call instance data.

5.6.3.1.2 Output

Inter-process communication

- POR – End (optional)

  Identifies the termination of a service process.

Inter-process data

None.

5.6.3.1.3 Graphical representation

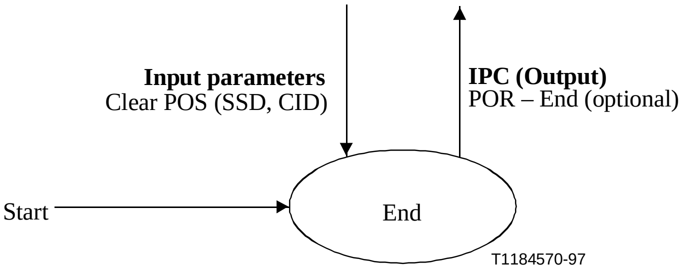


Figure 21/Q.1223 – End SIB operation

5.6.4 SIB graphical representation

END SIB

– End

Figure 22/Q.1223 – END SIB

5.7 INITIATE SERVICE PROCESS

5.7.1 Definition

Causes the execution of a parallel service process to begin.

5.7.2 Potential application services

The following services are the potential services that use this capability:

- UPT;
- VPN;
- customized call routing (public and private).

5.7.3 Interface

The INITIATE SERVICE PROCESS SIB interface consists of the following SIB operation:

- Initiate service process, which causes the execution of a parallel service process to begin.

5.7.3.1 Initiate service process

This SIB operation starts a new service process to be executed in parallel by sending a POI with Interprocess Data (IPD) attached. The POI is sent asynchronously, i.e. after transmitting the POI, one of the outlets is chosen.

5.7.3.1.1 Input

Logical start

Indicates the logical start of the SIB operation.

Input data parameters

- Process Id

  Specifies the identity of the service process to be initiated. This parameter behaves as service support data, call instance data or as service instance data.

5.7.3.1.2 Output

Logical end

- Outlet success

  This outlet indicates that the indicated service process was successfully initiated.

- Outlet error

  This outlet indicates that the initiate operation caused an error. The reason of this error is indicated by the output parameter "error cause".

Output data parameters

- Error cause

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

  1) missing parameter;

  2) invalid parameter value defined.

  This parameter behaves as call instance data.

Inter-process communication

- POI – Initiate

  Identifies the initiation of a new service process with IPD attached.

Inter-process data

1) Process Ids

   Identifies the process Ids of the initiating service process and any other service process with which the initiated service process may have to communicate (e.g. via SEND-WAIT SIBs) during service logic execution.

2) Miscellaneous data (optional)

   Identifies the data (SSD, CID and SID) to be sent along as IPD.

Graphical representation

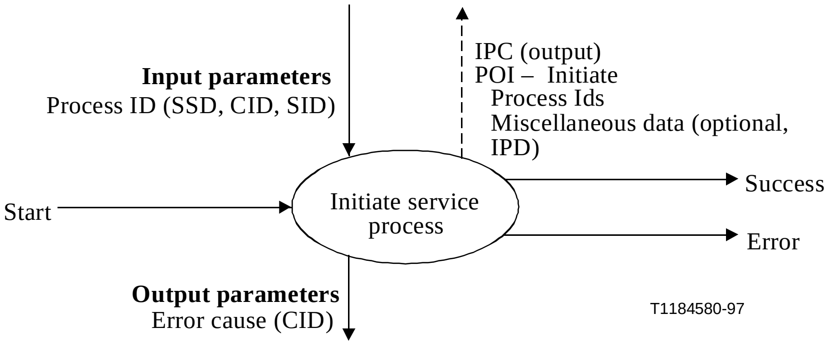


Figure 23/Q.1223 – Initiate service process SIB operation

5.7.4 SIB graphical representation

INITIATE SERVICE PROCESS SIB

– Initiate service process

Figure 24/Q.1223 – INITIATE SERVICE PROCESS SIB

5.8 JOIN

5.8.1 Definition

Attaches a call party or a group of call parties from the current call group into an indicated call group within the same call.

5.8.2 Potential application services

– call waiting;

– call hold with announcements;

– consultation calling;

– conference calling.

5.8.3 Interface

The JOIN SIB consists of one single SIB operation:

- Join call parties.

5.8.3.1 Join call parties

The Join call parties SIB operation attaches the indicated call party from the current call to an existing associated call group. The Join call parties SIB operation may also attach a whole group of call parties from the indicated call to existing call group within the same call.

5.8.3.1.1 Input

Logical start

Indicates the logical start of execution for this SIB operation.

Input data parameters

- Call parties

  Indicates the call party or group of call parties that has to be attached to indicated call group. This parameter behaves as service support data.

- Call group reference

  Indicates an existing call group where the indicated call parties have to be attached to. This parameter behaves as service support data.

Inter-process communication

None.

5.8.3.1.2 Output

Logical ends

- Outlet success

  This outlet indicates that the call parties are successfully attached to the call group.

- Outlet error

  This outlet indicates that the attachment of the call parties caused an error. The reason for this error is indicated by the output parameter "error cause".

Output data parameters

- Error cause

  Identifies the specific condition that caused an error during the operation of the SIB. The following error has been identified for the Join call parties SIB operation:

  – invalid call party.

  This parameter behaves as call instance data.

Inter-process communication

None.

5.8.3.1.3 Graphical representation

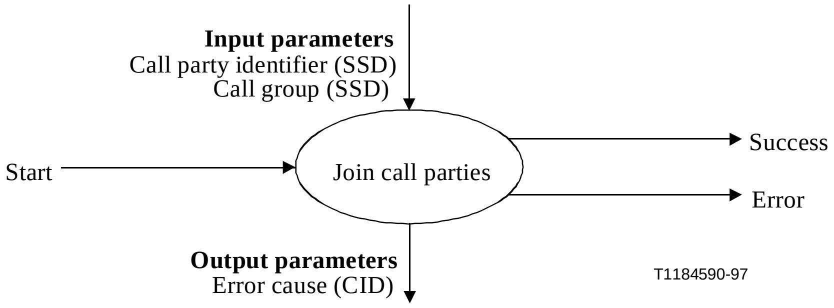


Figure 25/Q.1223 – Join call parties SIB operation

5.8.4 SIB graphical representation

JOIN SIB

– Join call parties

Figure 26/Q.1223 – JOIN 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 Potential application services

– All IN services.

5.9.3 Interface

The LOG CALL INFORMATION SIB consists of 2 SIB operations, log start and log end. Log start identifies the information to be logged and log end stores the identified data, once it becomes available. For information which is already available, it is preferable to use the data storage aspects of the SDM SIB.

Log start ⇨ Log end

5.9.3.1 Log start

This SIB operation identifies the information to be logged.

5.9.3.1.1 Input

Logical start

Indicates the logical start of the SIB operation.

Input data parameters

- Reference

  Specifies the reference to its succeeding log end SIB operation.

  This parameter behaves as service support data.

- Log

  Specifies which call instance data are to be logged. The following CID may be included:

  1) call attempt time;
  2) call stop time;
  3) call connect time;
  4) dialled number;
  5) destination number;
  6) additional dialled number (e.g. credit card number, etc.);
  7) calling line identification;
  8) time in queue;
  9) bearer capability;
  10) error causes;
  11) any other CID.

  This parameter behaves as call instance data.

Inter-process communication

None.

5.9.3.1.2 Output

Logical end

- Outlet success

  This outlet indicates that the logging was initiated successfully.

- Outlet error

  This outlet indicates that the log start operation caused an error. The reason of this error is indicated by the output parameter "error cause".

Output data parameters

- Error cause

  Identifies the specific condition that caused an error during the operation of the SIB. The following error has been identified for the Log start operation:

  – invalid log specified.

  This parameter behaves as call instance data.

5.9.3.1.3 Graphical representation

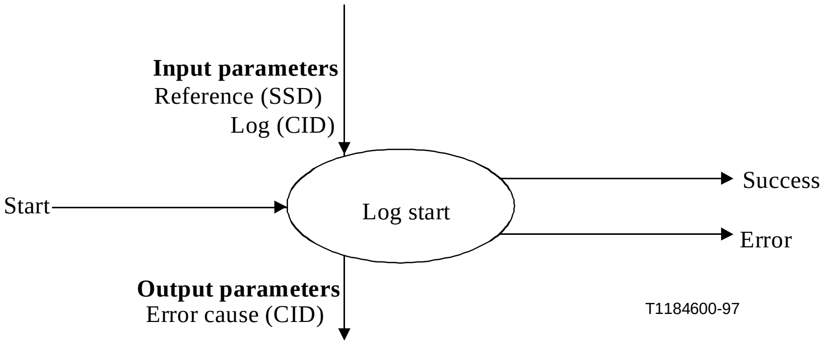


Figure 27/Q.1223 – Log start SIB operation

5.9.3.2 Log end

This SIB operation stores the identified data, once it becomes available.

5.9.3.2.1 Input

Logical start

Indicates the logical start of the SIB operation.

Input data parameters

- Reference

  Specifies the reference to its preceding Log start SIB operation.

  This parameter behaves as service support data.

- Log file indicator

  Specifies the log file where the values of the log is to be logged.

  This parameter behaves as call instance data.

- Authorized relationship Id

  Specifies the identity of the established authorized relationship through which operations can be applied.

Inter-process communication

None.

5.9.3.2.2 Output

Logical end

- Outlet success

  This outlet indicates that the log information has been stored.

- Outlet error

  This outlet indicates that an error was encountered.

Output data parameters

- Error cause

  Identifies the specific condition that caused an error during the operation of the SIB. The following error has been identified for the Log end operation:

  – invalid log file identifier.

  This parameter behaves as call instance data.

5.9.3.2.3 Graphical representation

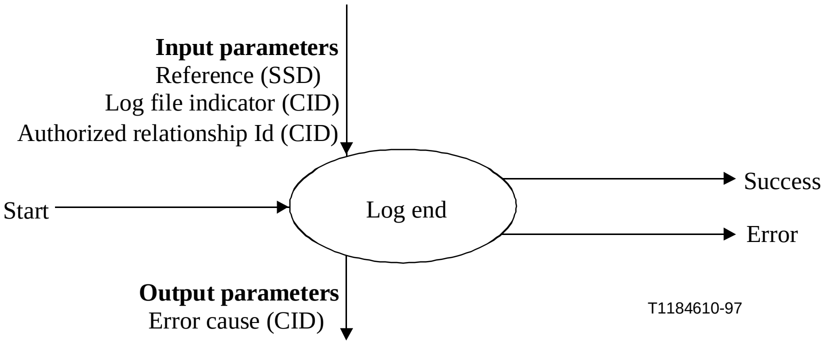


Figure 28/Q.1223 – Log end SIB operation

5.9.4 SIB graphical representation

LOG CALL INFORMATION

– Log start

– Log end

Figure 29/Q.1223 – LOG CALL INFORMATION SIB

5.10 MESSAGE HANDLER

5.10.1 Description

Sends a message conveyed with Inter-Process Data (IPD) between a controlling and a supporting service process.

5.10.2 Potential application services

The following services are the potential services that uses this capability:

- UPT;
- VPN;
- customized call routing (public and private).

5.10.3 Interface

The MESSAGE HANDLER SIB addresses two different modes of information exchange between controlling and supporting service processes (i.e. asynchronous and synchronous information exchange).

The MESSAGE HANDLER SIB consists of the following SIB operations:

- Send SIB operation: sends a message between a controlling and a supporting service process. Both directions are addressed.
- Receive SIB operation: processes a received message from another service process and, if not available, it will wait for the receipt of the indicated messages.

The stage 2 SDL representation of this SIB may be helpful in understanding its operation. Refer to 11.2.10/Q.1224.

5.10.3.1 Send message SIB operation

This SIB operation sends a POS with IPD attached to another service process that is executing in parallel. It shall be used for inter-process communication between parallel executing service processes. The SIB operation within asynchronous mode transmits one POS in an asynchronous communication event, i.e. the sending side will wait for the POS to be correctly buffered by the receiving side, but it will not wait for the receiving side to further process the POS with the attached IPD.

The Send message SIB operation within the synchronous mode transmits a POS in an synchronous communication event, i.e. the sending side will wait for the returning POS before it continues service logic processing.

To be able to send a POS to a particular service process, the identifier of that service process has to be known, i.e. a certain instance of a service process. This service process identifier is made available by the INITIATE SERVICE PROCESS SIB.

5.10.3.1.1 Input

Logical start

Indicates the logical start of execution for this SIB operation.

Input data parameters

- Process Id

  Identifies the identity of the receiving service process. This identifier can be specified global, local or user specific. This parameter behaves as service support data, call instance data or service instance data.

- Mode

  Identifies the mode of the information exchange (i.e. asynchronous or synchronous message exchange) and the class of the service processes (i.e. controlling or supporting service processes).

- Message types

  Identifies the message classes that are sent to the other service process. The following message classes are supported by the controlling service process.

  1) Information request

     This message request additional information from the supporting service process.

2) Notification event

This message reports the occurrence of requested events to the supporting service process. A confirmation on the receipt of this message can be requested.

3) Charging event

This message reports the requested charging information to the supporting service process. A confirmation on the receipt of this message can be requested.

The following message classes are supported by the supporting service process:

1) User information request

This message requests user interaction to the user provided by the controlling service process.

2) Charging information request

This message requests charging information of the call or user.

3) Event notification request

This message requests a notification on the occurrence of the specified events.

4) Information response

This message reports the requested information to the controlling service process.

Inter-process communication

None

5.10.3.1.2 Output

Logical end

- Outlet success

  This outlet indicates that transmission of the message to the other service process was successful and, if required, an acknowledgment of the transmitted message has been received.

- Outlet error

  This outlet indicates that the transmission of the message to the other service process caused an error. The reason of this error is indicated by the output parameter "error cause".

Output data parameters

- Error cause

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

  1) missing parameter;

  2) invalid parameter value defined;

  3) confirmation timeout occurred.

Inter-process communication

- POS – Send information

  Identifies the message sent to the other service process in the case of synchronous as well as asynchronous transmission.

  1) Sending service process Id

  Specifies the service process that transmits the "Send information" POS.

2) Message class

Specifies the message class that is transmitted to the other service process.

3) Miscellaneous data (optional)

Identifies the data that has to be sent.

5.10.3.1.3 Graphical representation

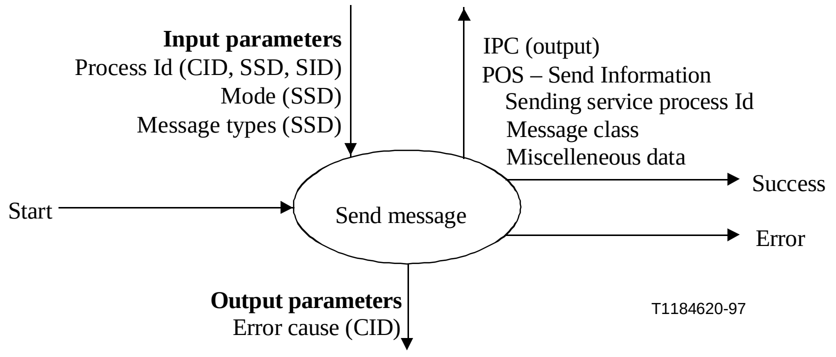


Figure 30/Q.1223 – Send message SIB operation

5.10.3.2 Receive message SIB operation

This SIB operation processes the received messages from the other service process or waits for the arrival of the messages if they are not yet received. The SIB operation fetches the first message within the message queue of its service process and processes the received messages by analyzing the POS delivered by the other service process and choosing a particular outlet of this SIB operation.

The Receive message SIB operation within the asynchronous mode will fetch a buffered POS and continues the service logic processing without acknowledgment to the service process that sent this particular POS.

The Receive message SIB operation within the synchronous mode will fetch a buffered POS and before continuing the service logic processing, it will provide an acknowledgment to the service process that sent this particular POS.

5.10.3.2.1 Input

Logical start

Indicates the logical start of execution for this SIB operation.

Input data parameters

- Process Id

  Identifies the identity of the sending service process. This identifier can be specified global, local or user specific. This parameter behaves as service support data, call instance data or service instance data.

- Mode

  Identifies the mode of the information exchange (i.e. asynchronous or synchronous message exchange) and the class of the service processes (i.e. controlling or supporting service processes).

- Message types

  Identifies the message classes that are received by the other service process. The following message classes are supported by the controlling service process:

  1) User information request

     This message requests user interaction to the user provided by the controlling service process.

  2) Charging information request

     This message requests charging information of the call or user.

  3) Event notification request

     This message requests a notification on the occurrence of the specified events.

  4) Information response

     This message reports the requested information to the controlling service process.

The following message classes are supported by the supporting service process:

1) Information request

   This message request additional information from the supporting service process.

2) Notification event

   This message reports the occurrence of requested events to the supporting service process. A confirmation on the receipt of this message can be requested.

3) Charging event

   This message reports the requested charging information to the supporting service process. A confirmation on the receipt of this message can be requested.

Inter-process communication

- POS – Receive message

  Identifies the received message from the other service process.

  1) Sending service process Id

     Specifies the service process that has transmitted the "Receive message" POS.

  2) Message class

     Specifies the message class that has been transmitted by the other service process.

  3) Miscellaneous data (optional)

     Identifies the data that has been received.

5.10.3.2.2 Output

Logical end

The following logical ends are supported in the case of controlling mode:

- Outlet user interaction (message type 1)

  This outlet indicates that additional user interaction is requested by the supporting service logic.

- Outlet notification requested (message type 3)

  This outlet indicates that notification of indicated event occurrences are required by the supporting service logic.

- Outlet charging requested (message type 2)

  This outlet indicates that charging information is requested by the supporting service logic.

- Outlet information report (message type 4)

  This outlet indicates that the requested information is received from the supporting service logic.

- Outlet error

  This outlet indicates that the receipt of the IPC messages caused an error. The reason of this error is indicated by the output data parameter "error cause"

The following logical ends are identified in case of the supporting mode:

- Outlet information report (message type 1)

  This outlet indicates that the controlling service logic requests for additional information.

- Outlet charging report (message type 3)

  This outlet indicates that the controlling service logic reports the requested charging information.

- Outlet notification report (message type 2)

  This outlet indicates that the controlling service logic reports the requested notification of the occurrence of the specified event.

- Outlet error

  This outlet indicates that the receipt of the IPC messages caused an error. The reason of this error is indicated by the output data parameter "error cause".

Output data parameters

- Error cause

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

  1) missing parameter;

  2) invalid parameter value defined;

  3) confirmation timeout occurred.

- Sending service process Id

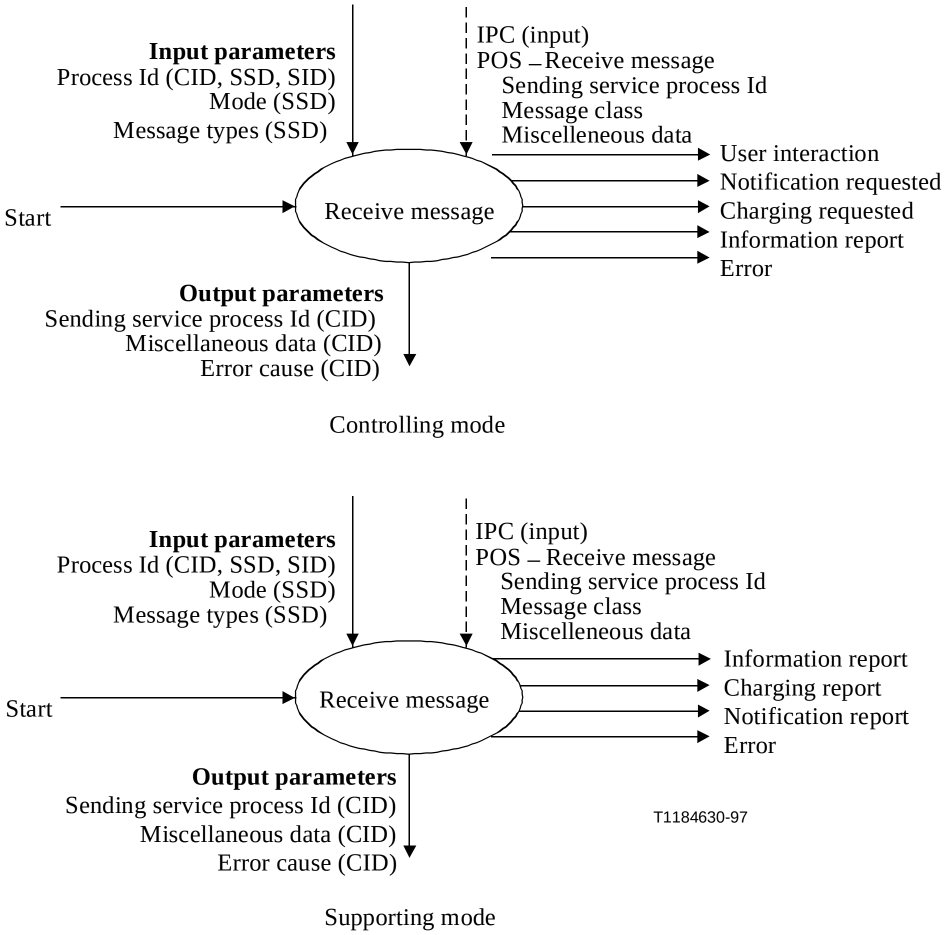
  Identifies the service process that has send the POS.

- Miscellaneous data (optional)

  Identifies the data to be received as IPD.

Inter-process communication

None.



Controlling mode

Supporting mode

Figure 31/Q.1223 – Receive message SIB operation

5.10.4 SIB graphical representation

MESSAGE HANDLER SIB

– Send message

– Receive message

Figure 32/Q.1223 – MESSAGE HANDLER SIB

5.11 QUEUE

5.11.1 Definition

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

5.11.2 Potential application services

The following services are the potential services that uses this capability:

- All IN services which use call queuing service feature.

5.11.3 Interface

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

- queue call: queues the call;
- queue monitor: monitors the queue events and releases the call from the queue.

The causal relationship between the SIB operations are:

Queue call ⇨ Queue monitor.

This SIB support two different modes of queuing:

1) Queuing with announcements to the queued call:

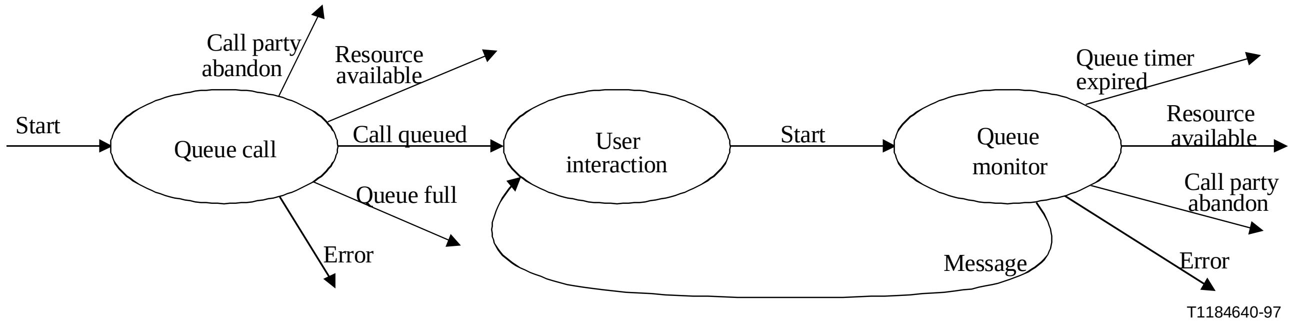


Figure 33/Q.1223 – Interaction of queue SIB operations with USER INTERACTION

NOTE – The description of the USER INTERACTION SIB is described in 5.18. It is assumed that queue monitor SIB operation will be succeeded by user interaction session close SIB operation.

2) Queuing without announcements to the queued call:

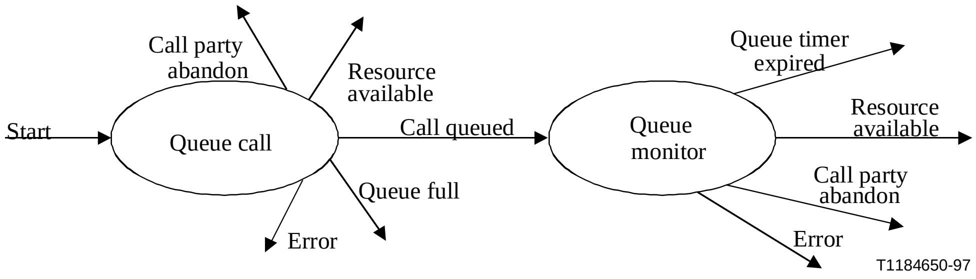


Figure 34/Q.1223 – Interactions of queue SIB operation without USER INTERACTION

5.11.3.1 Queue call

This SIB operation queues the call if no resource is available. If a resource is available, then the call will pass the queue and the "resource available" outlet will be chosen. During the queuing of the call, the queue will be checked for available queue space. If no queue space is available for the queuing of the call, then the outlet "queue full" will be chosen; else, the outlet "call queued" will be chosen. If an error occurs during the queuing of the call, then the outlet "error" will be selected. If the call party (i.e. caller) abandons the call, then the outlet "call party abandon" will be chosen.

If the announcement mode applies to this SIB operation, then specific queue information can be provided to the USER INTERACTION SIB, such as queue position of the queued call.

5.11.3.1.1 Input

Logical start

Indicates the logical start of the SIB operation.

Input data parameters

- Max active

  Specifies the maximum number of active calls allowed for the resource. This parameter behaves as SSD.

- Max number

  Specifies the maximum number of calls allowed on queue at a given time. This parameter behaves as SSD.

- Max time

  Specifies the maximum time the call may remain on the queue. This parameter behaves as SSD.

- Call reference

  Identifies the specific call which is a candidate for queuing. This parameter behaves as CID.

- Resource

  Specifies the data which identifies the resource for which the call will be queued. This parameter behaves as call instance data.

- Mode

  Specifies if announcements are played to the queued call. This parameter behaves as call instance data.

Inter-process communication

None.

5.11.3.1.2 Output

Logical end

- Outlet resource available

  This outlet indicates that the call is passed to the requested resource. No queuing was performed.

- Outlet queue full

  This outlet indicates that the queue resource rejected the call because no free queue position was available.

- Outlet call queued

  This outlet indicates that the call was queued.

- Outlet call party abandon

  This outlet indicates that caller has abandoned the service request.

- Outlet error

  This outlet indicates that the Queue operation caused an error. The reason of this error is indicated by the output parameter "error cause".

Output data parameters

- Error cause

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

  1) invalid max active;
  2) invalid max number;
  3) invalid max time;
  4) invalid call reference.

  This parameter behaves as call instance data.

- Message Id (announcement mode)

  Identifies the specific announcement information provided to the USER INTERACTION SIB (e.g. queue position number). This parameter is only relevant if the SIB operation operates in announcement mode. This parameter behaves as call instance data.

5.11.3.1.3 Graphical representation

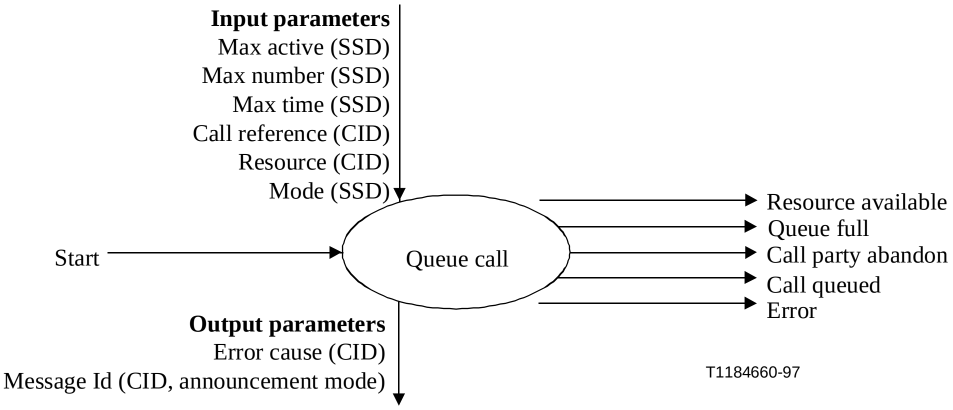


Figure 35/Q.1223 – Queue call SIB operation

5.11.3.2 Queue monitor

This SIB operation has two different modes of queuing, queuing with playing announcements to the queued call and queuing without announcements to the called queue. In case announcements are required, this SIB operation interacts with the USER INTERACTION SIB.

This SIB operation monitors all queue events of a call, previously queued by the Queue call SIB operation. The following queue events are identified:

- Changed queue position (announcement mode)

  This event indicates the fact that the call has been moved in the queue.

- Expired announcement timer (announcement mode)

  This event indicates the fact that repeated announcement timer has been expired and a new announcement has to be restarted.

- Expired queue timer

  This event indicates the fact that the maximum time of a call spent in a queue has been expired. In this case, the call will be released from the queue.

- Abandoned call party

  This event indicates the fact that the caller abandons the call. In this case, the call will be released from the queue.

- Free resource

  This event indicates the fact that a resource is available for the call. The call will be released from the queue.

This SIB operation dequeues the call if a resource becomes available. In this case, the outlet "resource available" will be chosen. If during the queuing of the call the queue timer expires, then the call will be forced to be dequeued and the outlet "queue timer expire" will be chosen. If the caller abandons the service request, then the outlet "call party abandon" will be selected and the call will be dequeued. If an active announcement is associated to the dequeued call, then the announcement will be cancelled.

If this SIB operation operates in announcement mode and the events "expired announcement timer" and "changed queue position" have occurred, then the SIB operation will select the "message" outlet.

5.11.3.2.1 Input

Logical start

Indicates the logical start of the SIB operation.

Input data parameters

- Call reference

  Identifies the specific call which is a candidate for queuing. This parameter behaves as call instance data.

- Resource

  Specifies the data which identifies the resource for which the call will be queued. This parameter behaves as call instance data.

- Message Id (announcement mode)

  Identifies the specific announcement information provided to the USER INTERACTION SIB (e.g. queue position number) by the queue call SIB operation. This parameter is only relevant if the SIB operation operates in announcement mode. This parameter behaves as call instance data.

- Announcement repetition timer (announcement mode)

  Identifies the repetition time of the announcement that are played to the queued call. This parameter is only relevant if the SIB operation operates in announcement mode. This parameter behaves as call instance data.

Inter-process communication

None.

Output

Logical end

- Outlet resource available

  This outlet indicates that the call is passed to the requested resource. No queuing was performed.

- Outlet call party abandon

  This outlet indicates that caller has abandoned the service request.

- Outlet queue timer expire

  This outlet indicates that the queue timer has expired and call has been dequeued.

- Outlet message (announcement mode)

  This outlet indicates that an announcement could be played to the queued call. This outlet is only relevant if the SIB operation operates in the announcement mode.

- Outlet error

  This outlet indicates that the queue operation caused an error. The reason for this error is indicated by the output parameter "error cause".

Output data parameters

- Error cause

  Identifies the specific condition which caused an error during the operation of the SIB. The following error has been identified for Queue monitor:

  – invalid call reference.

  This parameter behaves as call instance data.

- Time spent in queue

  Identifies the total time that a particular call has queued. This parameter behaves as call instance data.

- Message Id (announcement mode)

  Identifies the specific announcement information provided to the USER INTERACTION SIB (e.g. queue position number). This parameter is only relevant if the SIB operation operates in announcement mode. This parameter behaves as call instance data.

5.11.3.2.2 Graphical representation

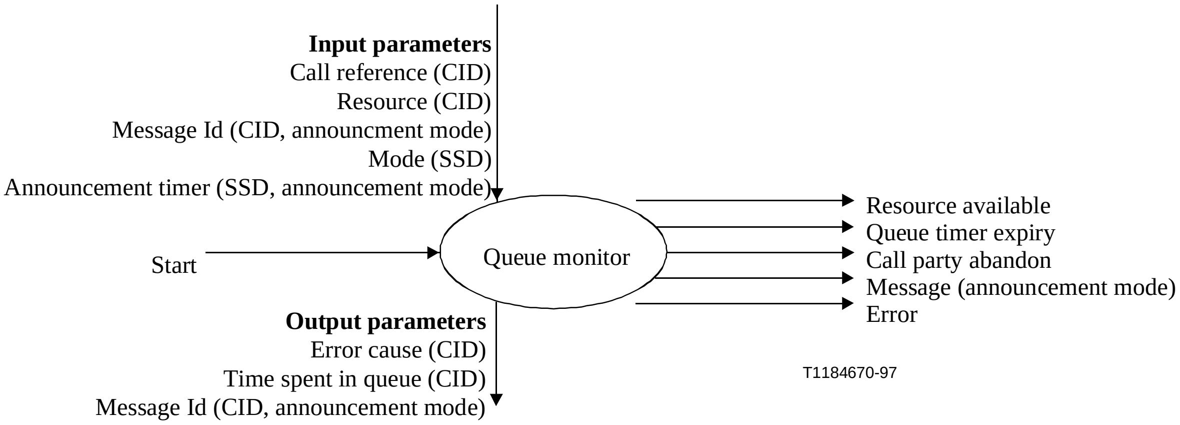


Figure 36/Q.1223 – Queue monitor SIB operation

5.11.4 SIB graphical representation

QUEUE SIB

– Queue call

– Queue monitor

Figure 37/Q.1223 – QUEUE SIB

5.12 SCREEN

5.12.1 Definition

Perform a comparison of data attributes against a filtered list of data attributes to determine whether the proposed values have been found in the list.

5.12.2 Potential service applications

- selective call forward on busy/no answer;
- originating call screening;
- terminating call screening;
- security screening;
- account card calling;
- credit card calling.

5.12.3 Interface

This SIB compares if a specified data attributes are contained within a specified list of data attributes.

The SCREEN SIB consists of one single SIB operation:

- SCREEN information.

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

Examples using the SCREEN SIB are:

- for verifying a user ID or PIN;
- for terminating or originating call screening based on a network address.

5.12.3.1 Screen information

This SIB operation takes the appropriate data attribute and determines whether or not it is contained within the list identified by the Service Support Data (SSD). A "match" condition results if the data attribute has been found on the list.

5.12.3.1.1 Input

Logical start

Indicates the logical start of execution for the SIB.

Input data parameters

- Screen list name

  The screen list name identifies the service data object to be used. This parameter behaves as a service support data or as call instance data.
- Screen list filter

  The screen list filter identifies the criteria for data attributes to be applied to the attributes in the screen data object to determine the set of list objects to be tested. This parameter behaves as service support data.
- Data

  The data parameter defines the data attributes which are to be tested against the selected list of service data objects. This parameter behaves as call instance data.
- Authorized relationship Id

  The authorized relationship Id provides the identity of the established authorized relationship through which the operations can be applied. This parameter behaves as call instance data.

Inter-process communication

None.

5.12.3.1.2 Output

Logical end

- Match [= on the list];
- No match;
- Error.

Output data parameters

- Error cause

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

  1) invalid identifier;

  2) invalid screen list.

Inter-process communication

None.

Graphical representation

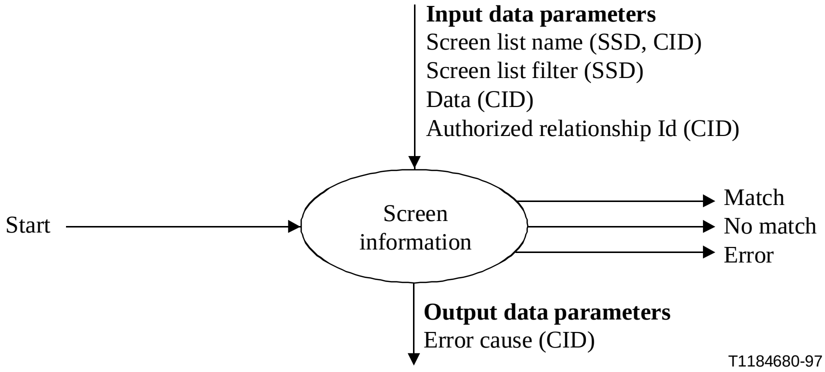


Figure 38/Q.1223 – Screen information SIB operation

5.12.4 SIB graphical representation

SCREEN SIB

– Screen information

Figure 39/Q.1223 – SCREEN SIB

5.13 SERVICE DATA MANAGEMENT

5.13.1 Definition

Enables action on service data (i.e. to be replaced, retrieved, incremented, decremented, stored and deleted).

5.13.2 Potential service applications

- call forwarding;
- customer profile management.

5.13.3 Interface

This SIB allows interaction with service data to add, modify or remove information from the service data.

The SERVICE DATA MANAGEMENT SIB consists of one single SIB operation:

- Service data management action.

5.13.3.1 Service data management action

This SIB operation performs the appropriate actions, i.e. replace, retrieve, increment, decrement, delete service data objects and service data attributes stored within the service data, execute a set of SDM actions on service data elements as an atomic action. For example, this SIB operation could be used to retrieve or replace a customer's call forwarding number.

5.13.3.1.1 Input

Logical start

Indicates the logical start of execution for the SIB operation.

Input data parameters

- Object name

  Specifies the service data object to be used. This parameter behaves as service support data or as call instance data.

- Action

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

  1) Replace – Replace the existing data attribute in the subscriber data object specified by the attribute indicator with new data specified by the information value.

  2) Retrieve – Copy the data attribute specified by the attribute indicator and place it in the output data parameter called retrieved value(s).

  3) Increment – Increase the value of the data attribute in the service data object specified by the attribute indicator by the amount indicated by the inc/dec value.

  4) Decrement – Decrease the value of the data attribute in the service data object specified by the attribute indicator by the amount indicated by the inc/dec value.

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

  6) 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.

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

  8) Execute – Execute a series of one or more actions, in a single operation. The operation is identified by the objectname and the execute identifier, and uses the input value. This encapsulation of a series of operations is known as an entry method.

  This parameter behaves as service support data.

- Attribute indicator

  Specifies the data attribute in the service data object upon which the action is to be performed. This parameter behaves as service support data or as call instance data.

- Attribute filter

  The attribute filter identifies the criteria for the data attributes to be applied to the attribute in the service data object to determine the set of list objects to be tested. This parameter behaves as service support data.

- Attribute filter value(s)

  The attribute filter value(s) defines the data attributes which are to be tested against the selected list of service data objects. This parameter behaves as call instance data.

- Inc/dec value

  Specifies the amount by which the attribute indicator is to be incremented or decremented. This parameter behaves as service support data.

- Authorized relationship Id

  The authorized relationship Id provides the identity of the established authorized relationship through which the operations can be applied. This parameter behaves as 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 data parameter called collected data (CID). This parameter behaves as call instance data.

- Execute identifier

  Specifies the entry method in the service data object which is to be executed. This parameter behaves as service support data.

- Input value

  Specifies the data to be used by the entry method. This parameter behaves as service support data or as call instance data.

5.13.3.1.2 Output

Logical end

- Success (record written or retrieved);
- Error.

Output data parameters

- Retrieved value(s)

  Specifies the data attribute(s) values retrieved. This parameter behaves as call instance data and is only relevant when the retrieve operation is selected.

- Execute result

  Specifies the data retrieved by the execution of an entry method. This parameter behaves as call instance data and is only relevant when the execute operation is selected.

- 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:

  1) invalid object name;
  2) invalid action;
  3) invalid attribute indicator;
  4) invalid information value;
  5) invalid inc/dec value;
  6) security error;
  7) invalid input value;
  8) execution error.

  This parameter behaves as call instance data.

Inter-process communication

None.

Graphical representation

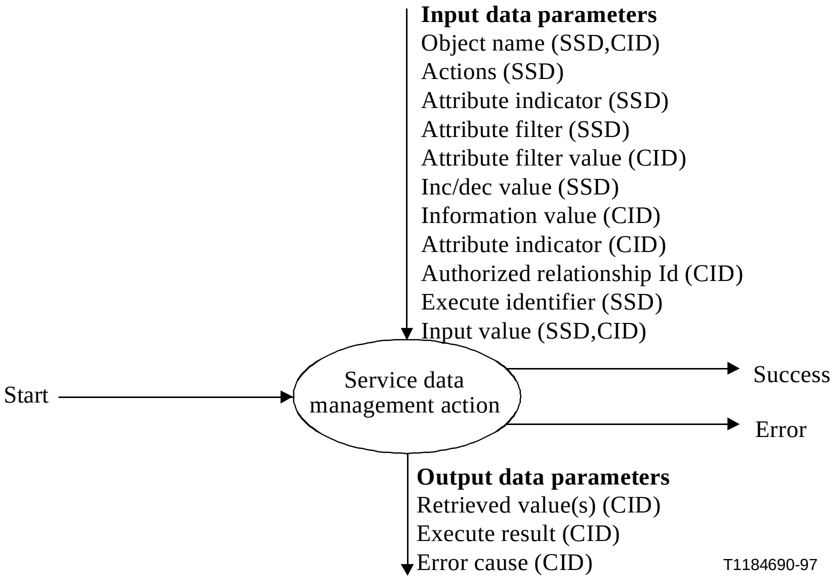


Figure 40/Q.1223 – Service data management action SIB operation

5.13.4 SIB graphical representation

SERVICE DATA MANAGEMENT SIB

– Service data managment action

Figure 41/Q.1223 – SERVICE DATA MANAGEMENT SIB

5.14 SERVICE FILTER

5.14.1 Definition

Filters the number of calls related to IN provided service features. Such filtering will be based on user specified parameters, such as service key, destination number. Filter response can be reported to the service logic.

5.14.2 Potential application services

The following services are the potential services that use this capability:

- (international) televoting;
- (international) mass calling.

5.14.3 Interface

This SIB is used to filter calls that are related to IN service features, and will specifically:

- filter IN invocations;
- provide statistical filter reports (optional).

The provisioning of filter reports is always preceded by IN service filtering. Therefore, the SIB operation "service filter reporting" will be preceded by the SIB operation "service filtering activating".

Service filter activating ⇨ service filter reporting.

NOTE – This SIB is used outside the context of a call, but must operate inside the context of a management activity. The invocation of such management logic is for further study.

5.14.3.1 Service filtering activating

This SIB operation is 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 following filter parameters can be specified:

- filter criteria;
- filter characteristics;
- treatment of filtered calls.

The Service filtering activating SIB operation in conjunction with other SIBs (e.g. COMPARE) can provide the required functionality for time dependent call limiting (e.g. time of day).

5.14.3.1.1 Input

Logical start

Indicates the logical start of the SIB operation.

Input data parameters

- Filter criteria

  Specifies the filtering criteria, such as service key, destination number, caller number and location number. A combination of these criteria is possible. This parameter conducts as service support data.

- Filter characteristics

  Specifies the severity of the filtering and frequency of the filtering. The two different filter mechanics are:

  1) Time interval

     After expiration of the interval timer, the next call will trigger the service logic.

  2) Call number interval

     After a specified number of calls, a call will be passed and service logic will be triggered.

  This parameter behaves as service support data.

- Filter timing

  Specifies the filtering duration. After the filtering expires, service filtering will be stopped and final statistics will be reported to the service logic. Two filtering timings are supported:

  1) Duration

     If the duration is expired, then service filtering will be stopped.

  2) Stop time

     If stop time is met, then service filtering will be stopped.

  This parameter behaves as service support data.

- Filtered call treatment

  Specifies how filtered calls are treated. It specifies the billing, announcement information, the maximum number of counters to be used and release information. This parameter behaves as service support data.

Inter-process communication

None.

5.14.3.1.2 Output

Logical end

- Outlet success

  This outlet indicates that the service filtering was initiated successfully.

- Outlet error

  This outlet indicates that the queue operation caused an error. The reason for this error is indicated by the output parameter "error cause".

Output data parameters

- Error cause

  Identifies the specific condition that caused an error during the operation of the SIB. The following errors have been identified for the user interaction play SIB operation:

  1) invalid filter criteria specified;

  2) invalid filtered call treatment specified.

  This parameter behaves as call instance data.

5.14.3.1.3 Graphical representation

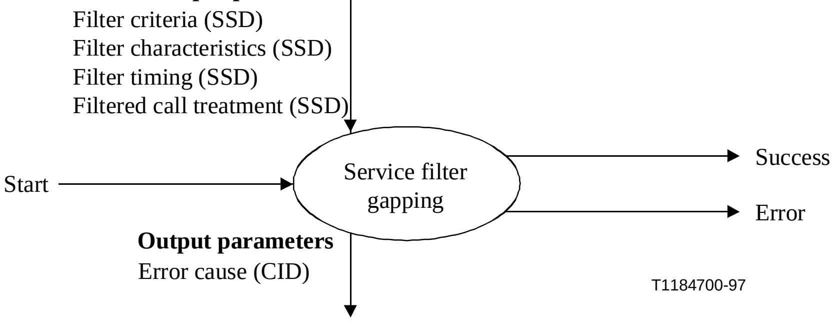


Figure 42/Q.1223 – Service filter activating SIB operation

5.14.3.2 Service filtering reporting

This SIB operation provides the service logic with statistical information on the filtered calls. Upon receipt of this information, the outlet "report" will be selected. If the service filtering has been stopped, then the outlet "end" will be selected to indicate the final statistical reporting.

5.14.3.2.1 Input

Logical start

Indicates the logical start of the SIB operation.

Input data parameters

None.

Inter-process communication

None.

5.14.3.2.2 Output

Logical end

- Outlet report

  This outlet indicates that statistical information has been received.

- Outlet end

  This outlet indicates that the final statistical information has been received and service filtering has ended.

Output data parameters

- Statistical report

  Specifies the statistics reports of the filtered calls. Each counter is related to a specific called party number. The number of counters are specified by the input parameter "filtered call treatment".

  This parameter behaves as call instance data.

5.14.3.2.3 Graphical representation

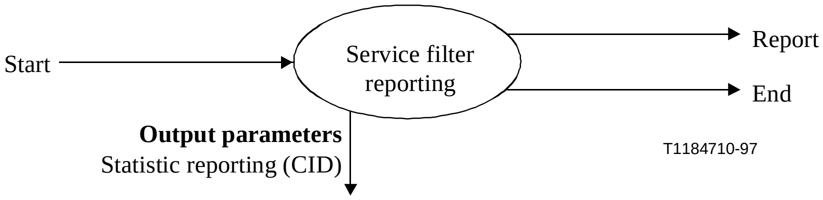


Figure 43/Q.1223 – Service filter reporting SIB operation

5.14.4 SIB graphical representation

SERVICE FILTER SIB

– Service filter activating

– Service filter reporting

Figure 44/Q.1223 – SERVICE FILTER SIB

5.15 SPLIT

5.15.1 Definition

Detaches a call party or a group of call parties from the current call and attaches the indicated call parties in new initiated call or another existing call.

5.15.2 Potential application services

- call waiting;
- call hold with announcements;
- consultation calling;
- conference calling.

5.15.3 Interface

The SPLIT SIB consists of one single SIB operation:

- split call parties.

5.15.3.1 Split call parties

The Split call parties SIB operation detaches the indicated call party from the current call. It will attach the detached call party to a new associated call group or to an existing associated call group. The Split call parties SIB operation may also detach a whole group of call parties from the indicated call and it attaches the call group to a new created call or existing call.

5.15.3.1.1 Input

Logical start

Indicates the logical start of execution for this SIB operation.

Input data parameters

- Call parties

  Indicates the call party or group of call parties that has to be disconnected from the call or call group. This parameter behaves as service support data.

- Call reference 2 (optional)

  Indicates an existing call reference or an associated call group where the call parties have to be moved to. This parameter behaves as service support data.

Inter-process communication

None.

5.15.3.1.2 Output

Logical ends

- Outlet success

  This outlet indicates that the call parties are successfully detached from the call.

- Outlet error

  This outlet indicates that the detachment of the call parties caused an error. The reason for this error is indicated by the output parameter "error cause".

Output data parameters

- New call reference

  Specifies the call that contains the detached call parties. This parameter behaves as call instance data.

- Error cause

  Identifies the specific condition that caused an error during the operation of the SIB. The following error has been identified for the Split call parties SIB operation:

  – invalid call party

  This parameter behaves as call instance data.

Inter-process communication

None.

5.15.3.1.3 Graphical representation

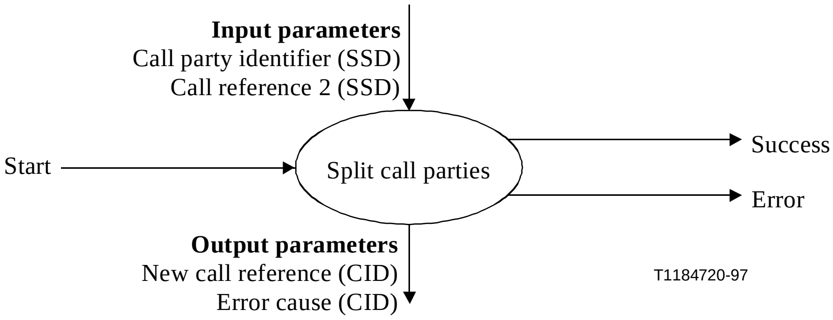


Figure 45/Q.1223 – Split call parties SIB operation

5.15.4 SIB graphical representation

SPLIT SIB

– Split call parties

Figure 46/Q.1223 – SPLIT SIB

5.16 STATUS NOTIFICATION

5.16.1 Definition

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

5.16.2 Potential service applications

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

5.16.3 Interface

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.

The STATUS NOTIFICATION SIB consists of 2 SIB operations: Status notification request and Status notification report. Status notification request identifies the information to be notified and status notification report reports the identified information, once it becomes applicable. For information which is already available, it is preferable to use the data storage aspects of the SDM SIB.

Status notification request ⇨ status notification report

5.16.3.1 Status notification request

This SIB operation identifies the information that has to be notified to the service logic.

5.16.3.1.1 Input

Logical start

Indicates the logical start of execution for the SIB.

Input data parameters

- Type

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

  1) poll resource status;
  2) wait for status;
  3) initiate continuous monitor;
  4) cancel continuous monitor.

  This parameter behaves as service support data.

- Resource

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

  - lines;
  - trunks.

  This parameter behaves as service support data

- Timer

  Specifies the maximum amount of time to monitor the resource.

  This parameter behaves as service support data.

- Resource status

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

  - busy;
  - idle.

  This parameter behaves as service support data.

Inter-process communication

None.

5.16.3.1.2 Output

Logical end

- Success;
- Error.

Output data parameter

- 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 request:

  a) invalid type;

  b) invalid resource;

  c) invalid timer;

  d) invalid resource status.

  This parameter behaves as call instance data.

Inter-process communication

None.

5.16.3.1.3 Graphical representation

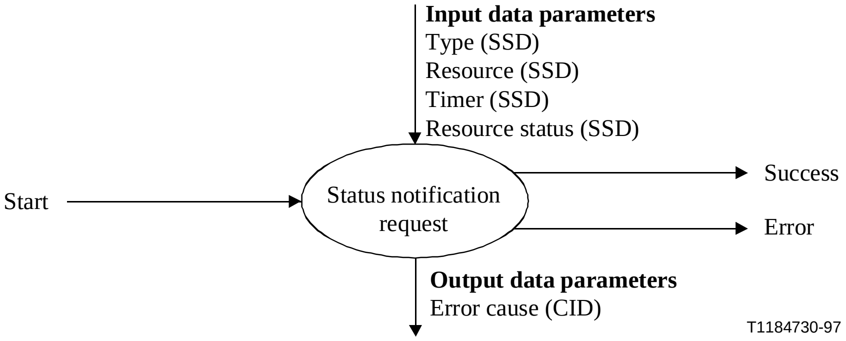


Figure 47/Q.1223 – Status notification request SIB operation

5.16.3.2 Status notification report

This SIB operation notifies the requested information to the service logic.

5.16.3.2.1 Input

Logical start

Indicates the logical start of execution for the SIB.

Input data parameters

- Status notification file

  Specifies the file object where the current resource status is to be logged.

  This parameter behaves as service support data.

Inter-process communication

None.

5.16.3.2.2 Output

Logical end

- Success;
- Timer expiry (only used with type "wait for status");
- Error.

Output data parameter

- Status

  Specifies the current status of the resource.

- Error cause

  Identifies the specific condition which caused an error during the operation of this SIB operation. The following error has been identified for Status notification report:

  – invalid status notification file indicator.

  This parameter behaves as call instance data.

Inter-process communication

None.

5.16.3.2.3 Graphical representation

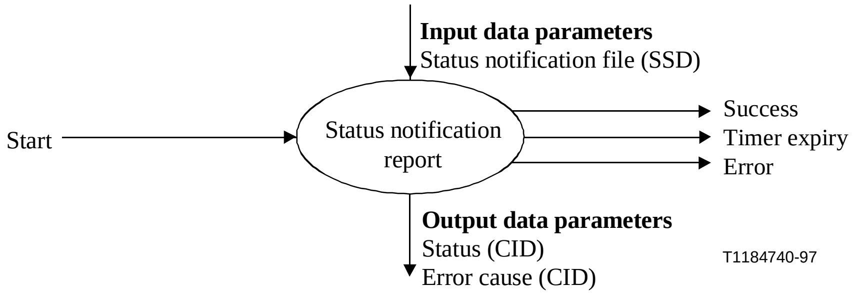


Figure 48/Q.1223 – Status notification report SIB operation

5.16.4 SIB graphical representation

STATUS NOTIFICATION SIB

– Status notification request

– Status notification report

Figure 49/Q.1223 – STATUS NOTIFICATION SIB

5.17 TRANSLATE

5.17.1 Definition

Determines output information from input information.

5.17.2 Potential service applications

- freephone;
- user-defined routing;
- VPN;
- UPT;
- abbreviated dialling;
- selective call forwarding on busy/no answer;
- call forwarding;
- call transfer.

5.17.3 Interface

This SIB translates input information to output information. The TRANSLATE SIB consists of one single SIB operation:

- translate data.

5.17.3.1 Translate data

This SIB operation 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 operation 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 Data SIB operation can provide the required functionality for time dependent routing.

5.17.3.1.1 Input

Logical start

Indicates the logical start of execution for the SIB.

Input data parameters

- Object name

  Specifies where the translation data object is located.

  This parameter behaves as service support data or as call instance data.

- 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 filter value(s) pointers.

  This parameter behaves as service support data.

- Translated attribute

  Specifies which object attribute(s) is to be returned to translated data values(s).

  This parameter behaves as service support data.

- Filter value(s)

  Specifies which attribute value are to be used as the information.

  This parameter behaves as service support data.

- Authorized relationship Id

  Specifies which data is to be used as the authorized relationship Id.

  This parameter behaves as call instance data.

Inter-process communication

None.

5.17.3.1.2 Output

Logical end

- Success;
- Error.

Output data parameters

- Translated data

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

  This parameter behaves as 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 TRANSLATE:

  a) invalid type;

  b) invalid file indicator;

  c) invalid information;

  d) translation not available.

  This parameter behaves as call instance data.

Inter-process communication

None.

5.17.3.1.3 Graphical representation

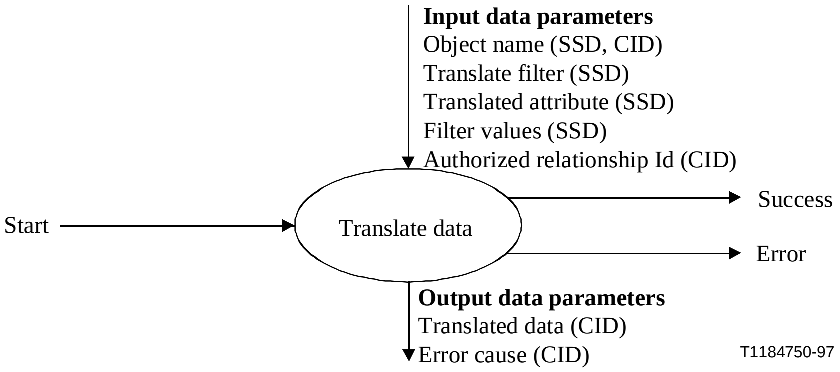


Figure 50/Q.1223 – Translate data SIB operation

5.17.4 SIB graphical representation

TRANSLATE SIB

– Translate data

Figure 51/Q.1223 – TRANSLATE SIB

5.18 USER INTERACTION

5.18.1 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.

5.18.2 Potential application services

The following services are the potential services that use this capability:

- Most IN CS-1 and CS-2 services will require user interaction.

5.18.3 Interface

This SIB provides all the processing needed to provide user interaction, and will specifically:

- start a user interaction session;
- play announcements to the user;
- play announcements to the user and collects information from the user;
- run a user interaction script;
- provide to the user interaction script additional information during its execution;
- close a user interaction script;
- end an user interaction session.

The SIB operations "User interaction session open" and "User interaction session close" are used to establish and release a network connection to the information platform, that provides the requested user interaction capability (i.e. announcements, messages, user interaction script).

The SIB operations "User interaction play" and "User interaction play and collect" are used independently but the SIB operation "user interaction play" does not allow interruptions of its played announcements and the SIB operation "play and collects" allows for interrupts of its announcements.

The SIB operations "User interaction run", "User interaction information" and "User interaction close" are required to precede in the following sequence:

User interaction run ⇒ user interaction information (zero or more) ⇒ user interaction close.

The stage 2 SDL representation of this SIB may be helpful in understanding its operation. Refer to 11.2.18/Q.1224.

5.18.3.1 User interaction play

This SIB operation provides a call party with information (e.g. announcements).

The announcements can be, for example:

- a customized or generic audio message;
- network progression tones (e.g. dial tone, busy tone, etc.);
- information messages (e.g. out-band information);
- service to user information (e.g. call waiting message).

This SIB operation 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.

The exchange of information messages is specified by the component data parameter.

5.18.3.1.1 Input

Logical start

Indicates the logical start of the SIB operation.

Input data parameters

- Mode

  Specifies the supported modes of information exchange capabilities (i.e. outband or inband) between the user and the network. The following modes are supported:

  1) outband (i.e. information messages);

2) inband (i.e. announcements, service information).

This parameter behaves as service support data or service instance data.

- Announcement ID

  Specifies which announcement is to be sent. This parameter behaves as service support data or as call instance data.

- Repetition requested

  Specifies if the announcement is to be repeated. This parameter behaves as service support data.

- Repetition interval

  Specifies the delay period in seconds between repetitions. This parameter behaves as service support data.

- Maximum repetitions

  Specifies the maximum number of times the announcement will be repeated. This parameter behaves as service support data.

- Duration

  Specifies the maximum length of time the announcement will be played. This parameter behaves as service support data.

- Call party identifier

  Specifies the data that identifies the call party receiving information from the network. This parameter behaves as call instance data.

- Component to be sent parameters

  Specifies the component to be sent to the end user. This parameter behaves as service support data or as call instance data

Inter-process communication

None.

5.18.3.1.2 Output

Logical end

- Outlet success

  This outlet indicates that the information has been provided successfully to the indicated call party.

- Outlet error

  This outlet indicates that the queue operation caused an error. The reason for this error is indicated by the output parameter "error cause".

Output data parameters

- Error cause

  Identifies the specific condition that caused an error during the operation of the SIB. The following errors have been identified for the user interaction play SIB operation:

  1) call/association abandon/released;

  2) announcement resource unavailable;

  3) invalid announcement ID;

  4) invalid call party;

5) call status incompatible with playing of announcements;

6) incompatible mode.

This parameter behaves as call instance data.

5.18.3.1.3 Graphical representation

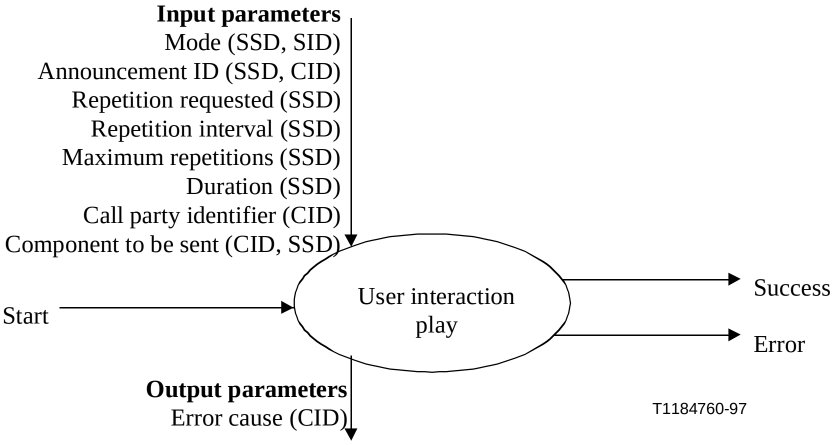

Input parameters
Mode (SSD, SID)
Announcement ID (SSD, CID)
Repetition requested (SSD)
Repetition interval (SSD)
Maximum repetitions (SSD)
Duration (SSD)
Call party identifier (CID)
Component to be sent (CID, SSD)
Start
User interaction play
Success
Error
Output parameters
Error cause (CID)
T1184760-97

Figure 52/Q.1223 – User interaction play SIB operation

5.18.3.2 User interaction play and collect

This SIB operation provides a call party with information and collects information from a call party.

The announcements can be, for example:

- a customized or generic audio message;
- network progression tones (e.g. dial tone, busy tone, etc.);
- information messages (e.g. outband information);
- service to user message (e.g. message waiting).

The collected information can be, for example:

- DTMF tones;
- audio;
- IA5 string text;
- information messages (e.g. outband information);
- user to service message (e.g. UPT registration).

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 exchange of information messages is specified by the component data parameter.

5.18.3.2.1 Input

Logical start

Indicates the logical start of the SIB operation.

Input data parameters

- Announcement parameters

  1) Announcement ID

     Specifies which announcement is to be sent. This parameter behaves as service support data or as call instance data.

  2) Repetition requested

     Specifies if the announcement is to be repeated. This parameter behaves as service support data.

  3) Repetition interval

     Specifies the delay period in seconds between repetitions. This parameter behaves as service support data.

  4) Maximum repetitions

     Specifies the maximum number of times the announcement will be repeated. This parameter behaves as service support data.

  5) Duration

     Specifies the maximum length of time the announcement will be played. This parameter behaves as service support data.

  6) Component to be sent

     Specifies the component to be sent to the end user. This parameter behaves as service support data or as call instance data.

- Collect information parameters

  1) User interruptibility

     Specifies if an announcement can be interrupted by the call party entering information (yes or no). This parameter behaves as service support data.

  2) Voice feedback

     Specifies if the user is given a vocalization of the user's input. This parameter behaves as service support data.

  3) Type

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

     - DTMF;
     - audio;
     - IA5 string.

     This parameter behaves as service support data.

  4) Maximum number of characters

     Specifies the maximum number of characters to collect, (> = 0, where 0 identifies non-character input). This parameter behaves as service support data.

5) Minimum number of characters

   Specifies the minimum number of characters to collect, (< = the maximum, > = 0, where 0 identifies non-character input). This parameter behaves as service support data.

6) Initial input waiting timer

   Specifies the maximum time to wait for beginning of call party's response. This parameter behaves as service support data.

7) Inter-character waiting timer

   Specifies the maximum time to wait after a pause by the call party. This parameter behaves as service support data.

8) End delineator

   Specifies special character(s) signifying the end of an input. If this parameter is "null", no delineator is specified. This parameter behaves as service support data.

9) 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 letters only;
   - a Lower case letters 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\ Specifies 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.

   This parameter behaves as service support data or as call instance data.

10) To be received component

   Specifies the component to be received by the network from the end-user. This parameter behaves as service support data or call instance data.

11) Call Party Identifier

   Specifies the data which identifies the call party receiving information from the network. This parameter behaves as call instance data.

- Sending mode

  Specifies the supported modes of information exchange capabilities (i.e. outband or inband) between the user and the network. The following modes are supported:

  – outband (i.e. information messages);

  – inband (i.e. announcements, service information).

  This parameter behaves as service support data or service instance data.

- Receiving mode

  Specifies the supported modes of information exchange capabilities (i.e. outband or inband) between the user and the network. The following modes are supported:

  – outband (i.e. information messages);

  – inband (i.e. DTMF, user information)

This parameter behaves as service support data or service instance data.

- Call party identifier

  Specifies the data that identifies the call party receiving information from the network. This parameter behaves as call instance data.

Inter-process communication

None.

5.18.3.2.2 Output

Logical end

- Outlet success

  This outlet indicates that the information has been provided successfully to the indicated call party.

- Outlet fail

  This outlet indicates that the information that has been received from the user had the incorrect syntax.

- Outlet error

  This outlet indicates that the queue operation caused an error. The reason for this error is indicated by the output parameter "error cause".

Output data parameters

- Collected data

  Specifies the data collected by the network from the call party. This parameter behaves as call instance data.

- Received component

  Specifies the component that was sent to the network by the end user. This parameter behaves as 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 user interaction collect SIB operation:

  1) call/association abandon/released;
  2) collection time out (no input received);
  3) incorrect number of digits received;
  4) data collection resource unavailable;
  5) invalid call party;
  6) inconsistent timer setting;
  7) call status incompatible with collecting information;
  8) invalid syntax format;
  9) incompatible mode;

  This parameter behaves as call instance data.

5.18.3.2.3 Graphical representation

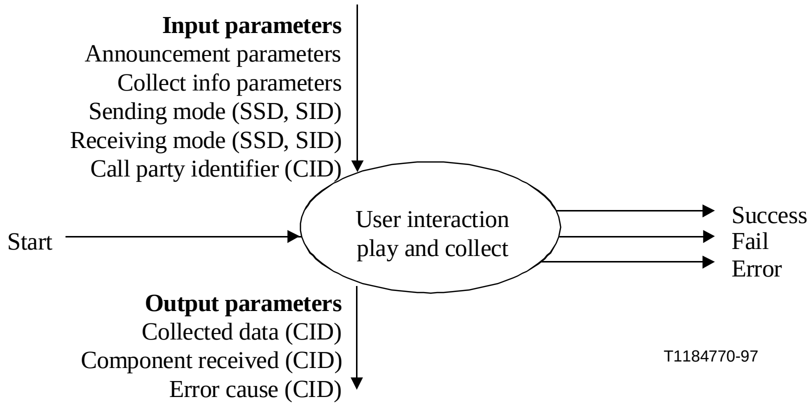


Figure 53/Q.1223 – User interaction play and collect SIB operation

5.18.3.3 User interaction run

This SIB operation provides a call party with a user interaction (possibly complex).

This SIB operation provides the specified user interaction script (possibly complex) to be played to the user. The exchange of information messages is specified by the component data parameter.

5.18.3.3.1 Input

Logical start

Indicates the logical start of the SIB operation.

Input data parameters

- User interaction script Id

  This parameter is used to identify the user interaction script. This parameter behaves as service support data.

- User interaction script specific information

  This parameter is used to provide information towards the invoked user interaction script. This parameter behaves as service support data.

Inter-process communication

None.

5.18.3.3.2 Output

Logical end

- Outlet success

  This outlet indicates that the user interaction script has been invoked successfully to the indicated call party.

- Outlet error

  This outlet indicates that the User interaction run operation caused an error. The reason for this error is indicated by the output parameter "error cause".

Output data parameters

- User interaction script result

  Returns the invoking service logic the result of the user interaction script. This result might be a partial result during the user interaction script execution or the final result of the user interaction.

  This parameter behaves as call instance data.

- Error cause

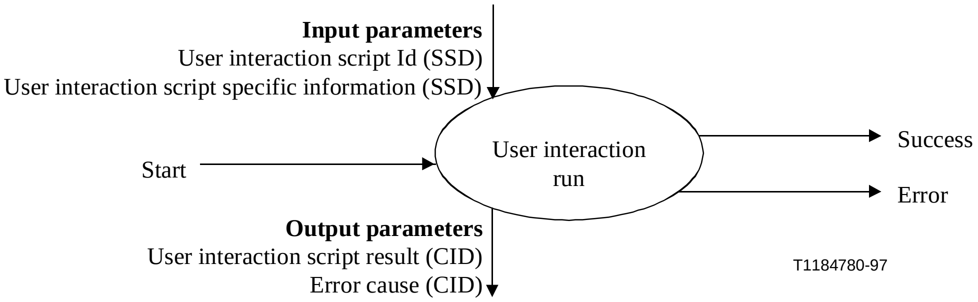
  Identifies the specific condition that caused an error during the operation of the SIB. The following errors have been identified for the User interaction run operation:

  1) invalid user interaction script Id;

  2) invalid user interaction information.

  This parameter behaves as call instance data.

5.18.3.3.3 Graphical representation



Input parameters
User interaction script Id (SSD)
User interaction script specific information (SSD)
Start
User interaction run
Success
Error
Output parameters
User interaction script result (CID)
Error cause (CID)
T1184780-97

Figure 54/Q.1223 – User interaction run SIB operation

5.18.3.4 User interaction information

This SIB operation provides the invoked user interaction script with additional information during its execution.

This SIB operation continues the specified user interaction script (possibly complex). The exchange of information messages is specified by the data parameter of this SIB operation.

5.18.3.4.1 Input

Logical start

Indicates the logical start of the SIB operation.

Input data parameters

- User interaction script Id

  This parameter is used to identify the user interaction script. This parameter behaves as service support data.

- User interaction script specific information

  This parameter is used to provide information towards the invoked user interaction script. This parameter behaves as service support data.

Inter-process communication

None.

5.18.3.4.2 Output

Logical end

- Outlet success

  This outlet indicates that the user interaction script has been played successfully to the indicated call party.

- Outlet error

  This outlet indicates that the user interaction information operation caused an error. The reason of this error is indicated by the output parameter "error cause".

Output data parameters

- User interaction script result

  Returns to the invoking service logic the result of the user interaction script. This result might be a partial result during the user interaction script execution or the final result of the user interaction.

  This parameter behaves as call instance data.

- Error cause

  Identifies the specific condition that caused an error during the operation of the SIB. The following errors have been identified for the User interaction run operation:

  1) invalid user interaction script Id;

  2) invalid user interaction information.

  This parameter behaves as call instance data.

5.18.3.4.3 Graphical representation

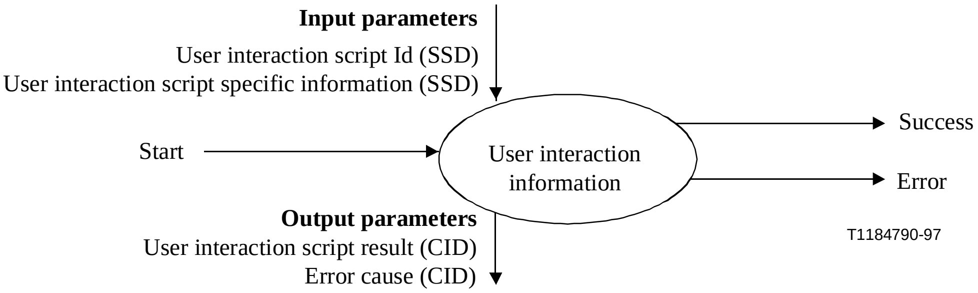

Input parameters
User interaction script Id (SSD)
User interaction script specific information (SSD)
Start
User interaction information
Success
Error
Output parameters
User interaction script result (CID)
Error cause (CID)
T1184790-97

Figure 55/Q.1223 – User interaction information SIB operation

5.18.3.5 User interaction close

This SIB operation ends the specified user interaction script. The exchange of information messages is specified by the data parameters of this SIB operation.

5.18.3.5.1 Input

Logical start

Indicates the logical start of the SIB operation.

Input data parameters

- User interaction script Id

  This parameter is used to identify the user interaction script. This parameter behaves as service support data.

- User interaction script specific information (optional)

  This parameter is used to provide information towards the invoked user interaction script. This parameter behaves as service support data.

Inter-process communication

None.

5.18.3.5.2 Output

Logical end

- Outlet success

  This outlet indicates that the user interaction script has been played successfully to the indicated call party.

- Outlet error

  This outlet indicates that the User interaction close operation caused an error. The reason of this error is indicated by the output parameter "error cause".

Output data parameters

- Error cause

  Identifies the specific condition that caused an error during the operation of the SIB. The following errors have been identified for the User interaction run operation:

  1) invalid user interaction script Id;

  2) invalid user interaction information.

  This parameter behaves as call instance data.

5.18.3.5.3 Graphical representation

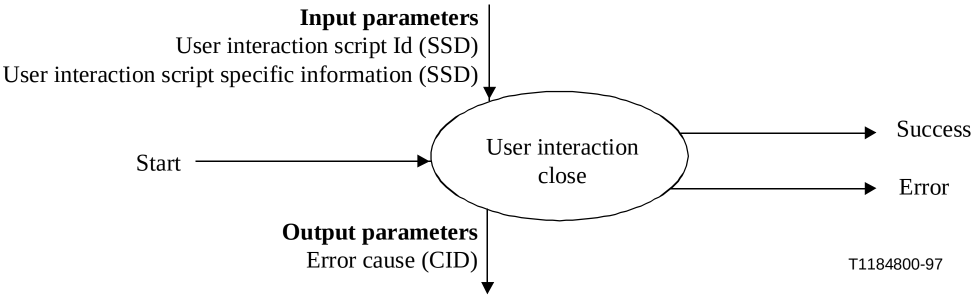


Figure 56/Q.1223 – User interaction close SIB operation

5.18.3.6 User interaction session open

This SIB operation initiates a user interaction session. It will establish a network connection to the information platform by establishing a speech path or a call unrelated association to this platform.

5.18.3.6.1 Input

Logical start

Indicates the logical start of the SIB operation.

Input data parameters

- Mode

  Specifies the requested modes of information exchange capabilities (i.e. outband or inband) between the user and the network. The following modes are supported:

  1) outband (i.e. information messages);

  2) inband (i.e. announcements);

  3) both modes.

  This parameter behaves as service support data.

- Call party identifier

  Specifies the data that identifies the call party receiving information from the network. This parameter behaves as call instance data.

Inter-process communication

None.

5.18.3.6.2 Output

Logical end

- Outlet success

  This outlet indicates that the network connection has been established successfully to the indicated call party.

- Outlet error

  This outlet indicates that the establishment of the network connection caused an error. The reason of this error is indicated by the output parameter "error cause".

Output data parameters

- Error cause

  Identifies the specific condition that caused an error during the operation of the SIB. The following errors have been identified for the User interaction session open SIB operation:

  1) invalid call party;

  2) call/association abandon/released.

  This parameter behaves as call instance data.

Inter-process communication

None.

5.18.3.6.3 Graphical representation

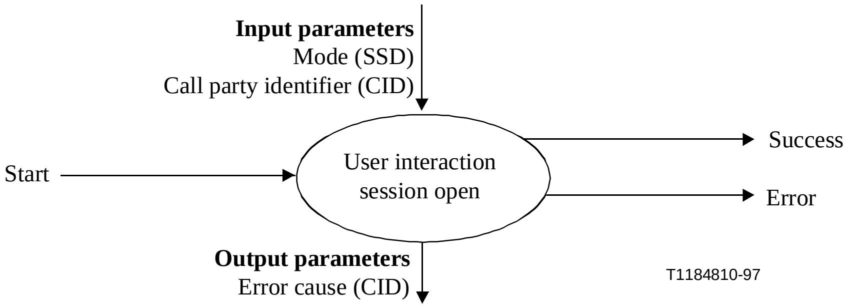


Figure 57/Q.1223 – User interaction session open SIB operation

5.18.3.7 User interaction session close

This SIB operation terminates a user interaction session. It will break the network connection to the information platform and release all the network resources.

5.18.3.7.1 Input

Logical start

Indicates the logical start of the SIB operation.

Input data parameters

- Call party identifier

  Specifies the data that identifies the call party receiving information from the network. This parameter behaves as call instance data.

Inter-process communication

None.

5.18.3.7.2 Output

Logical end

- Outlet success

  This outlet indicates that the network connection to the indicated call party has been released successfully.

- Outlet error

  This outlet indicates that the release of the network connection caused an error. The reason of this error is indicated by the output parameter "error cause".

Output data parameters

- Error cause

  Identifies the specific condition that caused an error during the operation of the SIB. The following error has been identified for the User interaction session close SIB operation:

  – invalid call party.

  This parameter behaves as call instance data.

Inter-process communication

None.

5.18.3.7.3 Graphical representation

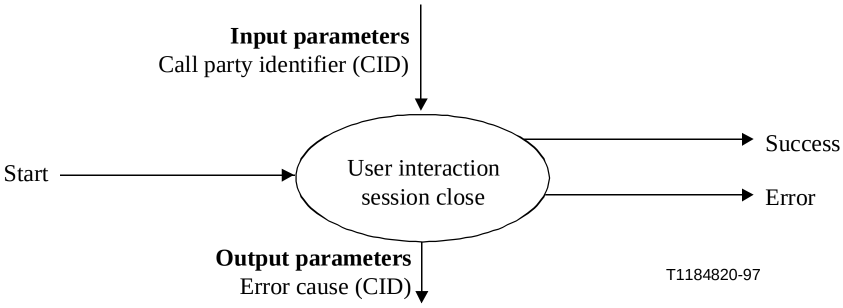


Figure 58/Q.1223 – User interaction session close SIB operation

5.18.4 SIB graphical representation

USER INTERACTION SIB

– User interaction session open

– User interaction play

– User interaction play and collect

– User interaction run

– User interaction information

– User interaction close

– User interaction session close

Figure 59/Q.1223 – USER INTERACTION SIB

5.19 VERIFY

5.19.1 Definition

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

5.19.2 Potential service applications

The following services are the potential services that use this capability:

- Any service that requires syntactical verification of user input.

5.19.3 Interface

This SIB compares call or user information with the format expected for the data. It consists of one single SIB operation:

- verify information.

5.19.3.1 Verify information

Information is compared with the format expected for the data. The verify information SIB operation 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 operation compares the input data to that expected format.

This SIB operation 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.

5.19.3.1.1 Input

Logical start

Indicates the logical start of the SIB operation.

Input data parameters

- Maximum number of characters

  Specifies the maximum number of characters allowed. This parameter behaves as service support data or as call instance data.
- Minimum number of characters

  Specifies the minimum of characters allowed. This parameter behaves as service support data or as call instance data.
- 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.

  This parameter behaves as service support data or as call instance data.
- Data

  Specifies the data which is to be verified. This parameter behaves as call instance data.

Inter-process communication

None.

5.19.3.1.2 Output

Logical end

- Outlet pass

  This outlet indicates that the verification of the call or user information was successful. The information has the expected format.

- Outlet fail

  This outlet indicates that the verification of the call or user information has failed. The information has not the expected format.

- Outlet error

  This outlet indicates that during the operation of the verification the error cause is indicated by the "error cause" output parameter.

Output data parameters

- Error cause

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

  1) invalid format;

  2) invalid identifier;

  3) inconsistent Max/Min number of characters specified.

Graphical representation

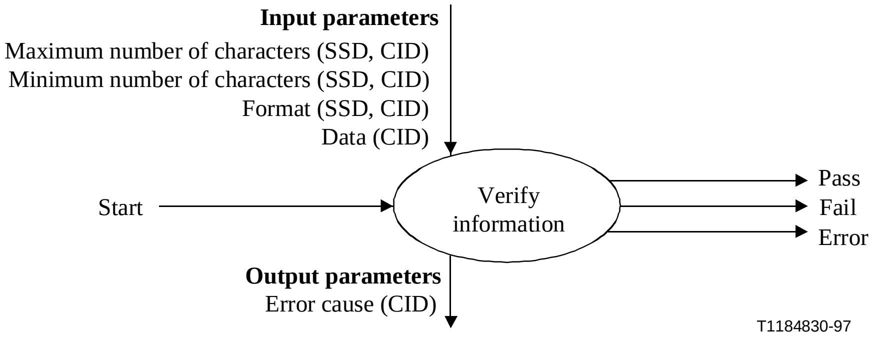


Figure 60/Q.1223 – Verify information SIB operation

5.19.4 SIB graphical representation

VERIFY SIB

– Verify information

Figure 61/Q.1223 – VERIFY SIB

6 IN CS-2 specialized SIBs

The following specialized SIBs are identified:

- Basic Call Process (BCP);
- Bearer Call Unrelated Process (BCUP).

6.1 The basic call process

6.1.1 General

The Basic Call Process (BCP) has been defined in clause 4/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 IN CS-2 services and service features.

It is not necessary or intended to fully model the BCP on the GFP in IN CS-2, 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-2 services/service features.

6.1.2 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 Synchronization (POS). These are defined in 4.2/I.329/Q.1203.

IN CS-2 service flexibility is provided by defining specific POI and POS 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 POS.

For example, an SIB chain which screens a network address against a list could represent an originating call screening service if the *address analyzed* 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 POS is used to complete the chain. If a "match" logical end of the SCREEN SIB is followed by a *proceed with new data* POS, 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* POS.

6.1.2.1 List of POIs

The following set of POIs has been identified for IN CS-2.

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).

Call authorized

This POI identifies that the user was authorized to make the service request without specifying a destination address.

Address collected

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

Address analyzed

This POI identifies that the address input has been analyzed 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.

Facility available

This POI identifies that the network has provided an available line or network resource to the user.

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.

Routing failure

This POI identifies that the network was not able to route the call to its destination.

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).

Call interrupted

This POI identifies that the user has interrupted the current call process to indicate a request for service handling.

Call suspended

This POI identifies that the calling party in case of originating service invocation has gone on-hook and re-answering of the call is still possible.

This POI identifies that the called party in case of terminating service invocation has gone on-hook and re-answering of the call is still possible.

Call re-answered

This POI identifies that the on-hook call party should be reconnected by alerting the call party.

End of call

This POI identifies that a call party has disconnected.

Call abandoned

This POI identifies that the call party has been abandoned by the call during the call set-up (i.e. originating side or terminating side).

6.1.2.2 List of POSs (PORs)

The following set of POSs (PORs) has been identified for IN CS-2.

Continue with existing data

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

Proceed with new data

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

Handle as transit

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

Release call party

This POS identifies that the BCP should release the indicated call party.

Release call groups

This POS identifies that the BCP should release the indicated call party group.

Reconnect

This POS identifies that the on-hook call party should be reconnected by alerting the call party.

Clear call

This POS identifies that the BCP should clear the call.

Initiate call

This POS 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.

6.1.3 BCP stage 1 description

6.1.3.1 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 POSs.

6.1.3.2 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 POS.

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

6.1.3.3 Potential service applications

- All IN CS-2 services, except for call unrelated services.

6.1.3.4 Output

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

6.1.3.4.1 Logical output

Initiates global service logic from a specified POI.

6.1.3.4.2 Service support data

- Set of POIs

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

6.1.3.4.3 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.

6.1.3.5 Input

6.1.3.5.1 Logical input

Resumes BCP at the point specified by the GSL specified POS.

6.1.3.5.2 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.

6.1.3.6 Graphical representation

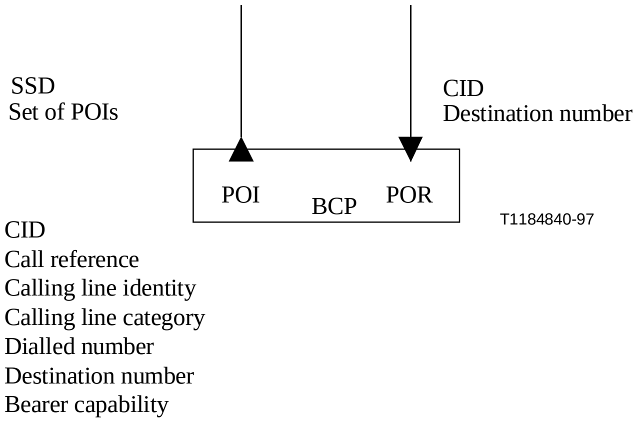


Figure 62/Q.1223 – Basic call process

6.2 Basic call unrelated process

6.2.1 General

The Basic Call Unrelated Process (BCUP) is defined as a specialized SIB which provides the call unrelated capabilities. These capabilities enable the use of GSL as well as other SIBs to completely describe IN CS-2 services and service features.

It is not necessary or intended to fully model the BCUP on the GFP in IN CS-2, but rather to insure that the functionality of the BCUP that is required on the GFP in conjunction with SIBs be available to fully support IN CS-2 services/service features.

6.2.2 Points of initiation and points of return

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

IN CS-2 service flexibility is provided by defining specific POI and POS interfaces between the BCUP and GSL. This flexibility allows a chain of SIBs to represent different services based on the launching POI and the returning POS.

For example, an SIB chain which authenticates a user or terminal could represent an authentication service if the *analyzed message* POI launches the chain.

Furthermore, using the screening example from above, different services can result based upon which POS is used to complete the chain. If a "success" logical end of the AUTHENTICATE SIB is followed by a *proceed with new data* POS, an association is allowed to continue and this is an example of successful authentication.

6.2.2.1 List of POIs

The following set of POIs has been identified for IN CS-2.

Analyzed message

This POI identifies that the message input has been analyzed to determine characteristics of the message (e.g. location updating).

Answer

This POI identifies that the interaction between the user and the network is active and the connection between the user and network is established.

Release association requested

This POI identifies that the call unrelated association is requested by the user to be released.

6.2.2.2 List of POSs (PORs)

The following set of POSs (PORs) has been identified for IN CS-2.

Continue with existing data

This POS identifies that the BCUP should continue processing with no modification.

Proceed with new data

This POS identifies that the BCUP should proceed with processing with only a data modification.

Release association

This POS identifies that the BCUP should release the association between the network and the user.

Initiate association

This POS identifies that the BCUP should initiate a call unrelated association between the network and the user.

6.2.3 BCUP stage 1 description

6.2.3.1 Definition

This specialized SIB allows access to IN services/service features. These services/service features can either be user initiated or network initiated and are realized outside the context of a call. The interface points between this SIB and GSL are described as POIs and POSs.

6.2.3.2 Operation

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

When the chain of SIB terminates, association processing may be influenced according to the specified POS.

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

6.2.3.3 Potential service applications

- user authentication;
- user registration;
- screening;
- call party interaction;
- activation/deactivation.

6.2.3.4 Output

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

6.2.3.4.1 Logical output

Initiates global service logic from a specified POI.

6.2.3.4.2 Service support data

- Set of POIs

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

6.2.3.4.3 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.).

- Invoked message

  Specifies the message originally invoked by the user.

- Message

  Specifies the message (the message, although originally the same as the invoked message, may be modified through IN service processing).

- Connection reference

  Identifies a specific connection.

6.2.3.5 Input

6.2.3.5.1 Logical input

Resumes BCUP at the point specified by the GSL specified POS.

6.2.3.5.2 Call instance data

- Message

  Specifies a message. For instance, a chain of SIBs may be used to create a message which differs from the invoked message.

6.2.3.6 Graphical representation

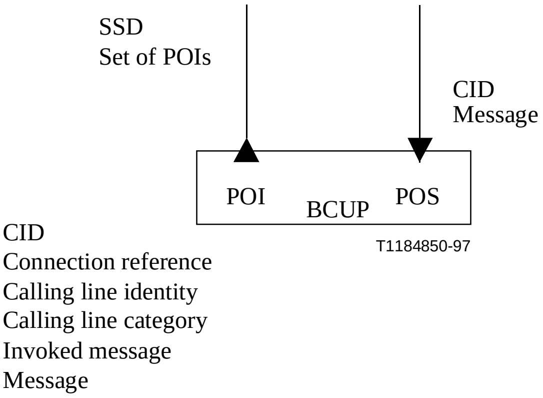
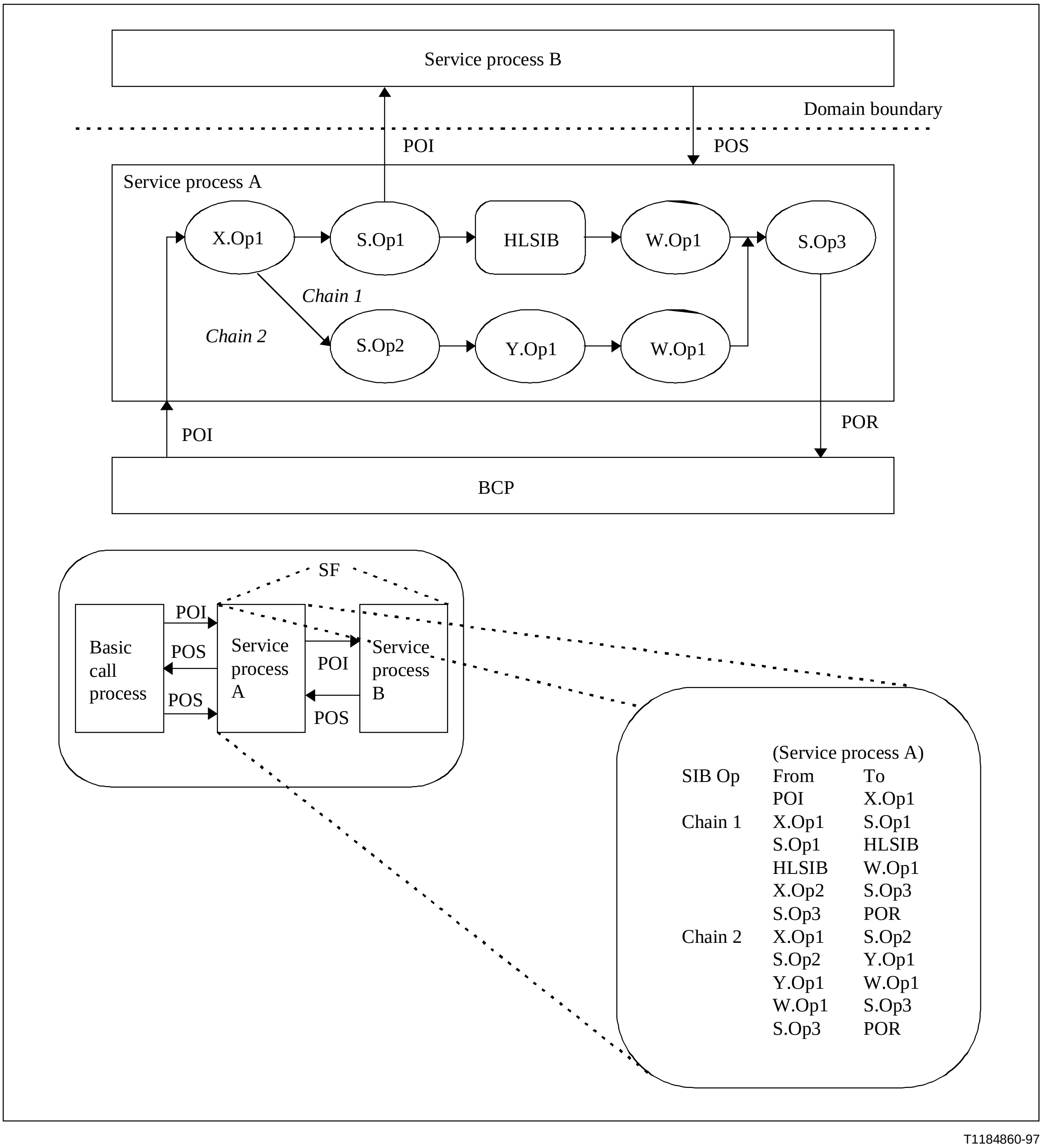


Figure 63/Q.1223 – Basic call unrelated process

7 Mapping of the service plane to the global functional plane

Referring to Figure 64 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 as a service process by a triggering mechanism from the basic call process. The chain "link" patterns for the service processes which describe the SF must then be obtained by the global service logic in order to process the SF. As new SFs are designed, their SIB operation descriptions must be made available to the global service logic.



	(Service process A)	
SIB Op	From	To
	POI	X.Op1
Chain 1	X.Op1	S.Op1
	S.Op1	HLSIB
	HLSIB	W.Op1
	X.Op2	S.Op3
	S.Op3	POR
Chain 2	X.Op1	S.Op2
	S.Op2	Y.Op1
	Y.Op1	W.Op1
	W.Op1	S.Op3
	S.Op3	POR

Figure 64/Q.1223 – Global functional plane blueprint of a service feature

ANNEX A

Overview of the SIBs and SIB operations

An overview of the SIBs and SIB operations is provided in Table A.1.

Table A.I/Q.1223 – Overview of the SIBs and SIB operations

SIB	SIB operation	Description
ALGORITHM Mass calling, televoting	Apply algorithm	Applies a mathematical algorithm to data to produce a data result.
AUTHENTICATE All services that access service data	Authenticate data user	Provides all the processing needed to establish an authorized relationship for a given user identity between service logic and service data.
	End authenticated relationship	Terminates an established authorized relationship.
CHARGE Any service with IN specific charging	Charge initiation	Used to initiate IN charging on any call requiring it.
	Charge reporting	Receives charging info and provides the service logic with a charge report.
COMPARE Time dependent routing, CCBS, credit card calling	Compare information	Compares an identifier to a specified reference value.
DISTRIBUTION Mass calling, televoting, freephone	Apply distribution	Distributes calls to its different logical ends based on a user identified algorithm.
LOG CALL INFORMATION All IN services	Log start	Identifies the information to be logged.
NOTE – The collected information may be used by management services (e.g. statistics) and not by call-related services.	Log end	Stores the identified data, once it becomes available. (For information which is already available, it is preferable to use the data storage aspects of the SDM SIB.)
QUEUE All IN services that use queuing	Queue call	Queues the call if no resource is available. Logical ends: "resource available", "call queued", "queue full" (no queue space is available),"call party abandon".
	Queue monitor	Dequeues the call if a resource becomes available. Logical ends: "resource available", "queue timer expire", "call party abandon", "message" (only if operated in announcement mode and the events "expired announcement timer" and "changed queue position" has been occurred).

Table A.I/Q.1223 – Overview of the SIBs and SIB operations *(continued)*

SIB	SIB operation	Description
SCREEN Selective call forward on busy/no answer, originating/terminating call screening, security screening, account card calling, credit card calling	Screen information	Takes the appropriate data attribute and determines whether or not it is contained within the list identified by the service support data (SSD). A "match" condition results if the data attribute has been found on the list.
SERVICE DATA MANAGEMENT Call forwarding, customer profile management	Service data management action	Performs the appropriate actions, i.e. replace, retrieve, increment, decrement, delete service data objects and service data attributes stored within the service data.
STATUS NOTIFICATION CCBS, call distribution, freephone, call transfer	Status notification request	Identifies the information that has to be notified to the service logic.
	Status notification report	Notifies the requested information to the service logic.
TRANSLATE Freephone, user-defined routing, VPN, UPT, abbreviated dialling, selective call forwarding on busy/no answer, call forwarding, call transfer	Translate data	Translates input information and provides output information, based on the various other input parameters.
USER INTERACTION All IN services that require user interaction	User interaction play	Provides a call party with information: audio message, network progression tones (e.g. dial tone, busy tone, etc.), information messages (e.g. outband information), service to user message (e.g. message waiting).
	User interaction play and collect	Provides a call party with information and collects information from a call party.
	User interaction run	Provides the specified user interaction script (possibly complex) to be played to the user.
	User interaction information	Provides the invoked user interaction script with additional information during its execution.
	User interaction close	Ends the specified user interaction script. The exchange of information messages is specified by the data parameters of this SIB operation.
	User interaction session open	Initiates a user interaction session. It will establish a network connection to the information platform by establishing a speech path or a call unrelated association to this platform.

Table A.I/Q.1223 – Overview of the SIBs and SIB operations *(continued)*

SIB	SIB operation	Description
	User interaction session close	Terminates a user interaction session. It will break the network connection to the information platform and release all the network resources.
VERIFY Any service that requires syntactical verification of (user) input	Verify information	Compares information with the format expected for the data. Follows normally the user interaction SIB when information has been collected from a call party.
JOIN Call waiting, call hold with announcement, consultation calling, conference calling	Join call parties	Attaches a call party or a group of call parties from the current call group into an indicated call group within the same call.
SERVICE FILTER (international) televoting, (international) mass calling NOTE – This SIB is used outside the context of a call, but must be operate inside the context of a management activity. The invocation of such management logic is described in Appendix I.	Service filtering activating	Used to pass all or a fraction of all calls related to IN provided service features.
	Service filtering reporting	Provides the service logic with statistical information on the filtered calls.
SPLIT Call waiting, call hold with announcement, consultation calling, conference calling	Split call parties	Detaches a call party or a group of call parties from the current call and attaches the indicated call parties in new initiated call or another existing call.
INITIATE SERVICE PROCESS UPT, VPN, customized call routing	Initiate service process	Starts a new service process to be executed in parallel by sending a POI with Interprocess Data (IPD) attached.
MESSAGE HANDLER UPT, VPN, customized call routing	Send message	Sends a message with interprocess data between a controlling and a supporting service process. Both directions are addressed. To be able to send a POS to a particular service process, the identifier of that service process has to be known, i.e. a certain instance of a service process. This service process identifier is made available by the INITIATE SERVICE PROCESS SIB.
	Receive message	Processes a received message from another service process and if not available it will wait for the receipt of the indicated messages.

Table A.I/Q.1223 – Overview of the SIBs and SIB operations *(concluded)*

SIB	SIB operation	Description
END UPT, VPN, customized call routing	End	Indicates the normal end of an executing service process or waits for requested events.
BASIC CALL PROCESS (BCP) All IN CS-2 services	Not defined	Provides the basic call capabilities.
BASIC CALL UNRELATED PROCESS (BCUP) User authentication, user registration, screening, call party interaction, activation/deactivation	Not defined	Provides the call unrelated capabilities.
BASIC SERVICE MANAGEMENT PROCESS (BSMP) All IN CS-2 service management services. The BSMP is described in Appendix I.	Not defined	Provides the service management capabilities.

APPENDIX I

IN management

I.1 Management view

This aspect of the GFP identifies the service management operations. These operations are contained within the SIBs and can be combined to reflect the management capabilities and facilitate the mapping of SIBs onto the Distributed Functional Plane (DFP) to help identify a comprehensive set for functional entity actions and information flows.

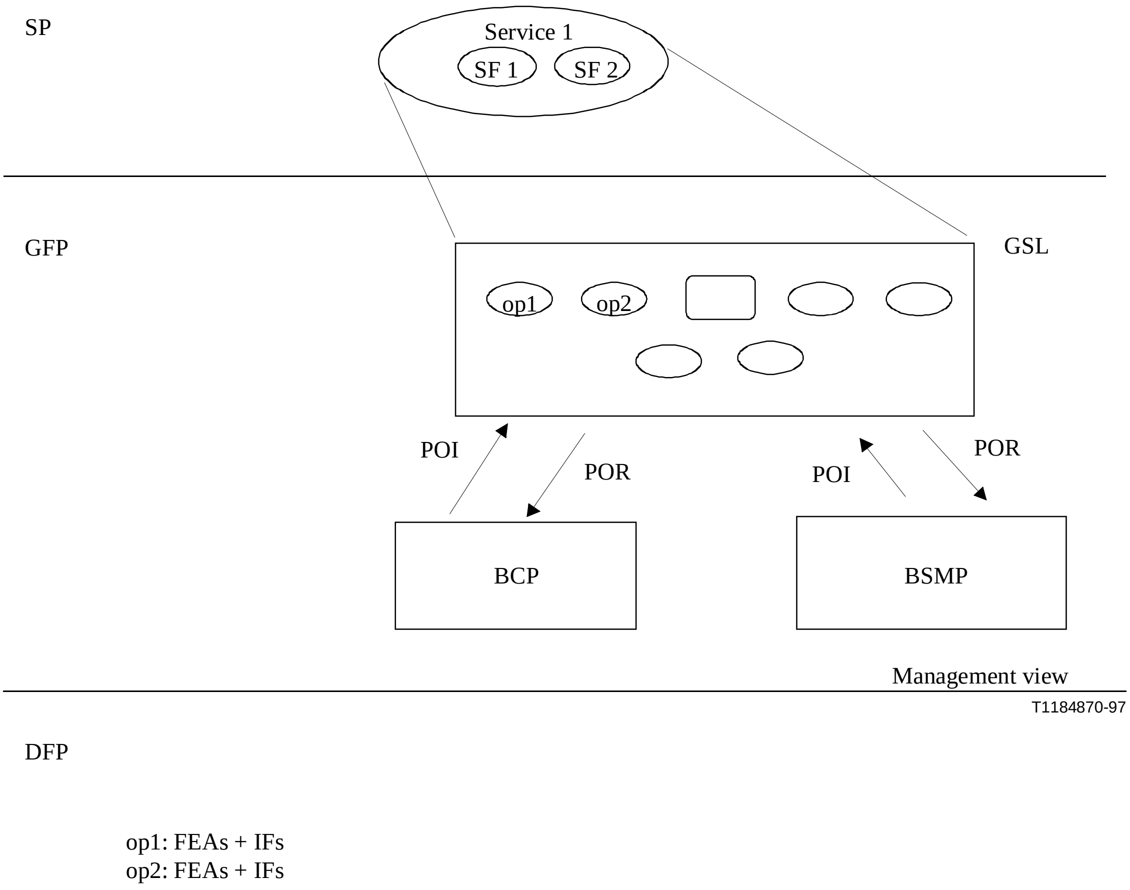


Figure I.1/Q.1223 – IN CS-2 GFP management view

This Appendix specifies how the model can be used to define management aspects in the GFP and aid the mapping of the GFP to the DFP.

The modelling elements within IN CS-2 GFP for management

1) BSMP – A specialized SIB which provides the basic service management capabilities.

2) Management process – A management process is considered as a combination of SIBs (containing SIB operations) or HLSIBs that represent the management activity.

   The service management aspects can be described in the GFP. Interaction points (POI, POS and POR) of management processes are included in the BSMP management process.

3) Management Instance Data (MID) – Management instance data defines dynamic parameters whose value will change with each management instance.

I.2 The Basic Service Management Process (BSMP)

The Basic Service Management Process (BSMP) is responsible for providing basic management functionality between parties in the network. The BSMP can be viewed as a specialized SIB, which provides a set of capabilities, such as:

- fault management;
- configuration management;
- performance management;
- security management;
- retaining POI/POS/MID/SSD for further processing.

The BSMP is described in more detail in this Appendix.

Characteristics of a basic service management process

A basic service management process represents a chain of SIBs or HLSIBs which are to be executed. This service management process is spawned via a POI and synchronized via POSs. Data can be communicated via MID or SSD.

Basic service management process has the following characteristics:

- Service management processes can only be composed out of HLSIBs and SIBs.
- Service management processes have only one entry point initiated via a POI.
- Communication between service management processes can be performed by POSs.
- Service management processes need mechanisms to send, receive and process MID/SSD via interaction points and to use the attached data. Data exchange is performed explicitly via interaction points.
- The basic mechanisms required are the creation of service management processes, and the interchange of data.
- Creation of service management processes: To create new service management processes, a specific mechanism has to be provided.
- Interchange of data: Service management processes may need to interact to synchronize and/or interchange data. A mechanism to support this requirement has to be provided.

IN CS-2 specialized SIBs processes

The following specialized SIB is identified:

– Basic Service Management Process (BSMP).

I.2.1 The basic service management process

General

The Basic Service Management Process (BSMP) is defined as a specialized SIB which provides the basic service management capabilities.

These capabilities enable the use of GSL as well as SIBs to completely describe IN CS-2 service management aspects.

It is not necessary or intended to fully model the BSMP on the GFP in IN CS-2, but rather to ensure that the functionality of the BSMP that is required on the GFP in conjunction with SIBs be available to fully support IN CS service management aspects.

The BSMP functional interfaces to the global service logic in the GFP are described as Points of Initiation (POI) and Points of Synchronization (POS).

IN CS-2 service management service flexibility is provided by defining specific POI and POS interfaces between the BSMP and GSL. This flexibility allows a chain of SIBs to represent different service management aspects based on the launching of POIs and the returning POSs.

BSMP stage 1 description

Definition

This specialized SIB allows access to IN service management aspects 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 POSs.

Operation

The BSMP contains a set of armed POIs, and if during the processing of an association, one of these POIs is encountered, a chain of SIBs is executed through global service logic.

When the chain of SIBs terminates, association processing may be influenced according to the specified POS.

Potential service applications

All IN CS-2 service management services.

Output

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

Logical output

Initiates global service logic from a specified POI.

Service support data/Management instance data

Set of POIs specifies the points in the BSMP.

ITU-T RECOMMENDATIONS SERIES

Series A	Organization of the work of the ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	**Switching and signalling**
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communication
Series Z	Programming languages