INTERNATIONAL TELECOMMUNICATION UNION

# CCITT

# I.329 / Q.1203

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

**INTEGRATED SERVICES**
**DIGITAL NETWORK (ISDN)**
**I.329** (10/92)

**OVERALL NETWORK ASPECTS**
**AND FUNCTIONS,**
**ISDN USER-NETWORK INTERFACES**
**Q.1203** (10/92)

---

# INTELLIGENT NETWORK – GLOBAL FUNCTIONAL PLANE ARCHITECTURE

**Recommendation I.329 / Q.1203**

---

## FOREWORD

The CCITT (the International Telegraph and Telephone Consultative Committee) is a permanent organ of the International Telecommunication Union (ITU). CCITT 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 Plenary Assembly of CCITT which meets every four years, establishes the topics for study and approves Recommendations prepared by its Study Groups. The approval of Recommendations by the members of CCITT between Plenary Assemblies is covered by the procedure laid down in CCITT Resolution No. 2 (Melbourne, 1988).

Recommendation I.329/Q.1203 was prepared by Study Group XVIII and was approved under the Resolution No. 2 procedure on the 1st of October 1992.

---

CCITT NOTES

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

2) A list of abbreviations used in this Recommendation can be found in Annex A.

## CONTENTS

**Recommendation I.329 / Q.1203**

# INTELLIGENT NETWORK – GLOBAL FUNCTIONAL PLANE ARCHITECTURE

*(1992)*

## 1 General

The concepts for the Intelligent Network (IN) are embodied in the IN Conceptual Model (INCM) as described in associated Recommendations I.312/Q.1201 and I.328/Q.1202. This section describes the global functional plane (GFP) of the INCM with respect to the composition of the plane, and its relationship to adjacent planes. This plane is viewed as the proper location for the modular functionality from which services are to be constructed.

The global functional plane models network functionality from a global, or network-wide, point of view. As such, the IN Structured Network is said to be viewed as a single entity in the GFP. In this plane, services and service features are redefined in terms of the broad network functions required to support them. These functions are neither service nor service feature (SF) specific and are referred to as service independent building blocks (SIBs).

The global functional plane is located between the service plane and the distributed functional plane as illustrated in Figure 1. Services identified in the service plane are decomposed into their service features then mapped onto one or more SIBs in the GFP. Each SIB is similarly mapped onto one or more Functional Entities in the Distributed Functional Plane.

Contained in the global functional plane are (refer to Figure 1):

– basic call process (BCP) SIB which identifies the normal call process from which IN services are launched, including points of initiation (POI) and points of return (POR) which provide the interface from the BCP to global service logic;

– SIBs which are standard reusable network-wide capabilities used to realize services and SFs;

– global service logic (GSL) which described how SIBs are chained together to describe service features. The GSL also describes interaction between the BCP and the SIB chains.

## 2 Global functional plane modelling

By definition, SIBs, including the BCP, are service independent and cannot contain knowledge of subsequent SIBs. Therefore, global service logic (GSL) is the only element in the GFP which is specifically service dependent.

Referring to the illustration in Figure 2 in the GFP, normal or non-IN supported services are processed within the BCP. When an IN supported service is to be invoked, its GSL is launched at the point of initiation (POI) by a triggering mechanism from the BCP.

In order to chain SIBs together, knowledge of the connection pattern, decision options, and data required by SIBs must be available. Therefore, the pattern of how SIBs are chained together must be maintained within the GFP, and described in the GSL. The GSL describes subsequential SIB chaining, potential branching, and where branches rejoin.

At the end of the chain of SIBs, the GSL also describes the returning point to the BCP by indicating the specific point of return (POR).

For a given service/SF, at least one POI is required. However, depending upon the logic required to support the service/SF, multiple PORs may be defined.

The process of how the GSL is described through the service creation environment using the application programming interface is an area for further study.

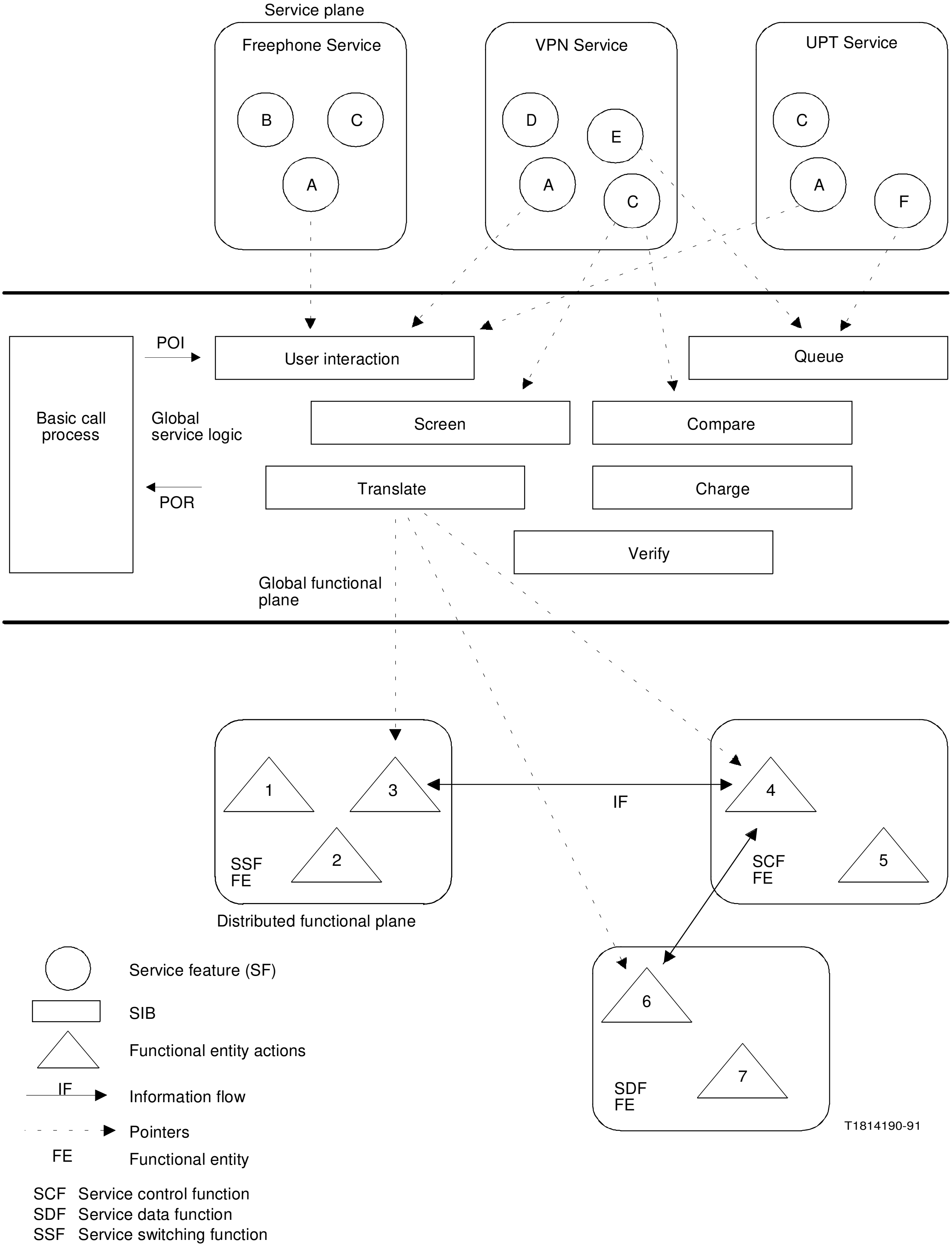


FIGURE 1

**Service decomposition**

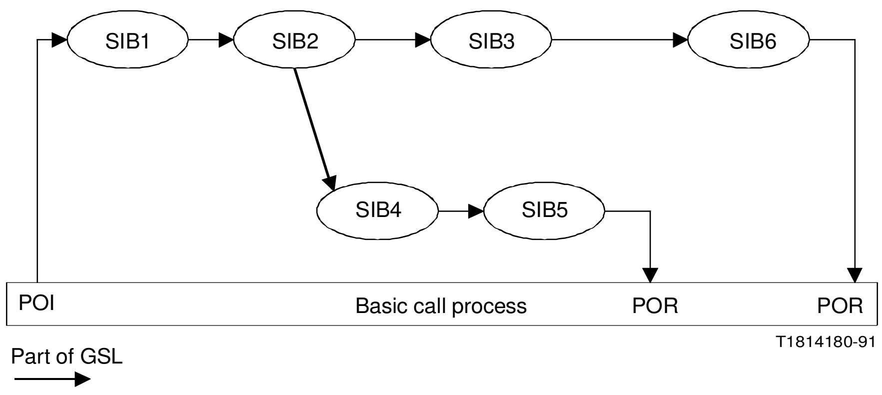


FIGURE 2

**Modelling of the global functional plane**

## 3 Service independent building blocks

### 3.1 *Definition of a* **SIB**

A 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 chained together in various combinations to realize services and SFs in the service plane. SIBs are defined to be independent of the specific service and technology for which or on which they will be realized.

### 3.2 *Characteristics of a 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.
- 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 a chain of SIBs.
- All SFs can be defined by a finite number of SIBs.

– A 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).

– A SIB has one logical starting point and one or more logical end points. Data required by each SIB is defined by SIB 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.

### 3.3 *Data parameters for SIBs*

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

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

Two types of data parameters are required for each SIB, dynamic parameters called call instance data (CID) and static parameters called service support data (SSD).

#### 3.3.1 *Call instance data (CID)*

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

– made available from the BCP SIB (e.g. Calling Line Identification);

– generated by a SIB, (e.g. a translated number); or

– entered by the subscriber, (e.g. a dialled number or a PIN code).

Associated with each CID value is a logical name which is referred to as the CID Field Pointer (CIDFP). If a SIB requires CID to perform its function, there will be an associated CIDFP assigned through SSD (refer to § 3.3.2). For instance, the Translate SIB's CID which defines what is to be translated is called Information. The Translate SIB's SSD parameter which defines where this data can be found is CIDFP-Info.

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

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

3.3.2 *Service support data (SSD)*

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

i) *Fixed parameters*

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

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

ii) *Field pointers*

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

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

3.4 *Method 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 (see Recommendation I.130), i.e. prose description, static description, and dynamic description, are appropriate.

The procedure outlined in Figure 4 can be used to determine if new SIBs are required to support new services.

The following terms are used in the SIBs identification method:

3.4.1 *Definition*

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

3.4.2 *Operation*

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

3.4.3 *Potential service applications*

Service examples of where this SIB can be used.

3.4.4 *Input*

Input to each SIB is specified as three distinct elements:

– one logical starting point;

– Service Support Data which defines parameters which are specified by the service description;

– Call Instance Data which are specific to that call instance.

3.4.5 *Output*

Output from each SIB is specified as two distinct elements:

– one or more logical end points;

– call instance data which defines data parameters specific to that call instance which results from the execution of that SIB and are required by other SIBs or the BCP to complete the call service instance.

3.4.6 *Graphic representation*

A graphic representation is used to describe input, operations and output of the SIB and is illustrated in Figure 3. Each SIB is characterized by having one logical input and one or more logical outputs. These logic flows are shown by the solid arrows on the left and right of the diagram. Each logic flow is specified above each arrow. SSD parameters are identified by the dashed arrow at the top of the diagram and are specified beside the dashed arrow. Similarly, CID parameters are specified below the diagram. Input CID parameters are separated from the output parameters.

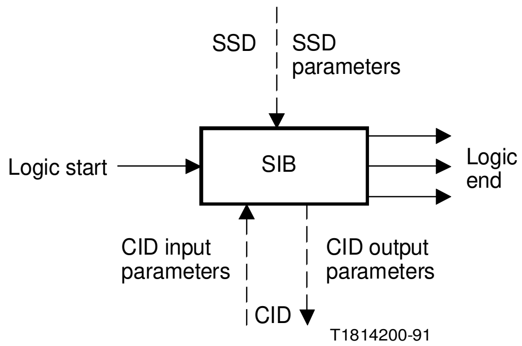


FIGURE 3

3.4.7 *SDL diagram*

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

3.5 *Flowchart analysis*

The starting point for the determination of SIBs is services. In the service plane of the IN Conceptual Model (INCM), services are decomposed into their Service Features (SF), which are the features that comprise the service. Full service descriptions must be available for the new service being analyzed prior to identifying SIBs.

Given that a catalogue of Services, SFs, and SIBs exist, the following description explains how analysis of a new service may lead to the extension of existing SIBs or identification of new SIBs (refer to Figure 4).

1) *List service features*

Decompose the new service into its SFs.

2) *Service feature definition*

Define each SF by describing the service provided from the end users (subscriber) perspective. This definition is referred to as the service prose. Information should be available from the stage 1 service description.

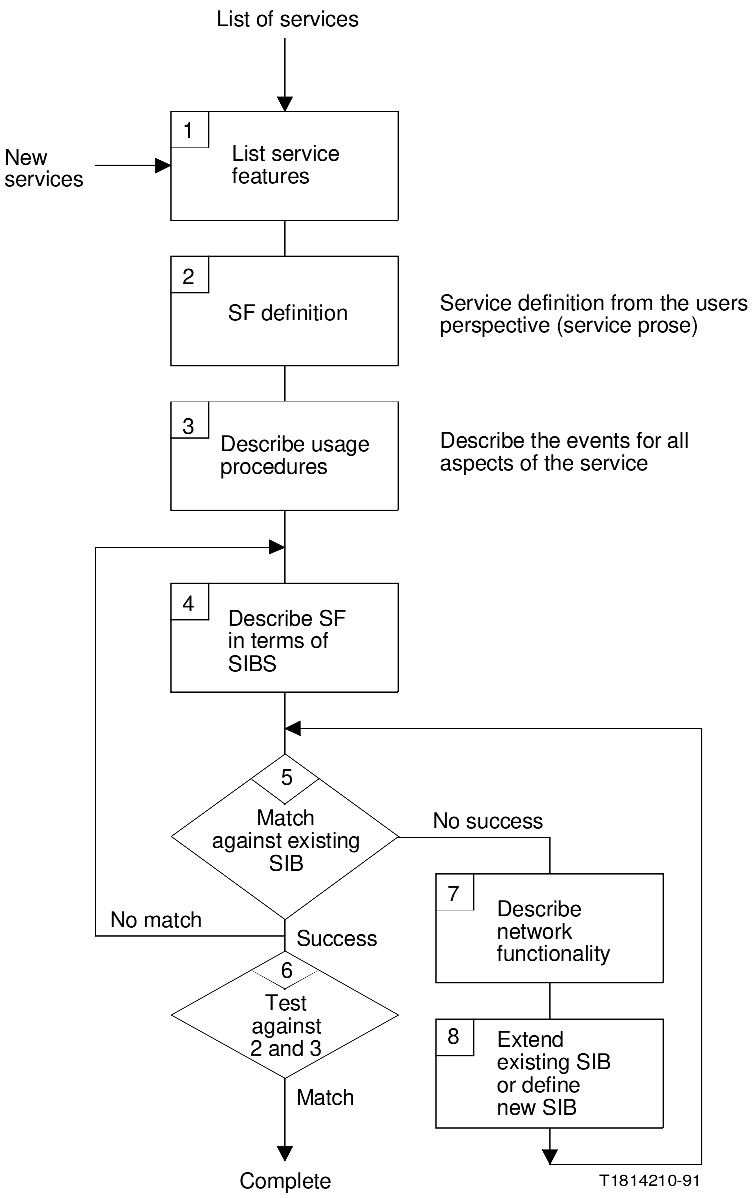


FIGURE 4

**Flowchart to identify service independent building blocks**

3) *Describe usage procedure*

Describe the chain of events seen by the user for this SF. This includes service subscription, activation, modification, and call scenarios for the SF.

4) *Describe service feature in terms of SIB*

Describe the SF in terms of the modular network functions represented by SIBs.

5) *Map to existing SIBs*

Compare the above (steps 3 and 4), with the characteristic lists for established SIBs.

6) *Test against 2 and 3*

Verify the robustness of the SF by analysing the SIB representation with the SF definition and usage procedure (from steps 2 and 3). Failure to pass this verification indicates that the analysis in steps 4 and 5 was incorrect or incomplete.

7) *Describe additional network functionality required*

Describe what functions must be provided by the network, in addition to those of existing SIBs, to fully support the SF.

8) *Extend existing SIB, or define new SIB*

If possible, extend the capabilities of an existing SIB (e.g. additional "type") to provide the additional functionality required to support the SF. If such extension is not possible, then define a new SIB. Complete the definition of the extended or new SIB by providing the detailed information in § 3.4.

# 4 Basic call process

## 4.1 *General*

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 then provides basic (e.g. Recommendation Q.17) call capabilities, including:

– connecting call, with appropriate disposition;

– disconnecting calls, with appropriate disposition;

– retaining CID for further processing of that call instance.

## 4.2 *Basic call process functionality*

IN supported services/SFs are represented through the use of chains of SIBs connected to the BCP SIB. The interface points between the BCP SIB and the chains of SIBs are described as points of initiation, and points of return, with the following definitions:

i) A **point of initiation** is the BCP functional launching point for the SIB chains.

ii) A **point of return** identifies the functional point in the BCP where the SIB chains terminate.

A graphical illustration of the POI/POR/BCP functionality is shown in Figure 5. The number and location of these points must be determined by analysis of the capabilities required for future capability sets.

The need for specific POI/POR functionality is that the same chain of SIBs may represent a different service if launched from a different point in the BCP. Similarly, the same chain of SIBs launched from the same point may represent a different service if returned to the BCP at a different point.

# 5 Global service logic

This section represents the role of the global service logic (GSL) in the global functional plane.

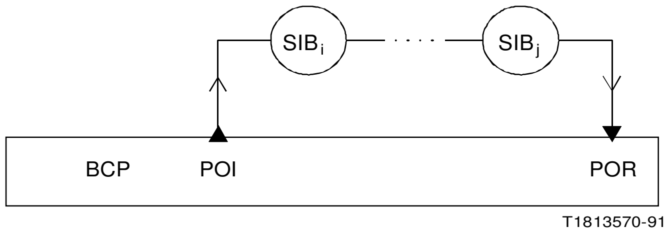


FIGURE 5
**POI/POR/BCP relationship model**

5.1 *General*

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

– BCP interaction points (POI and POR);

– SIBs;

– logical connections between SIBs, and between SIBs and BCP 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 6. 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.

5.2 *Relationship between GSL and BCP*

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

*Communications from BCP to GSL:*

i) logical start for SIB chains – which is represented by POIs;

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

*Communications from GSL to BCP:*

i) logical termination for SIB chains – which is represented by PORs;

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

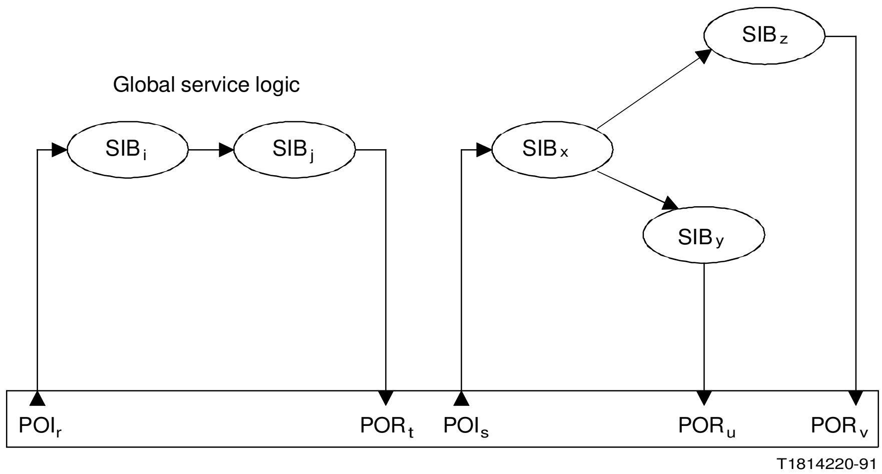


FIGURE 6
**GSL example**

5.3 *Relationship between global service logic and SIBs*

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

## 6 Mapping of the service plane to the global functional plane

Referring to Figure 7 in the GFP, non-IN services are processed through the basic call process. When an IN Service Feature (SF) is to be evoked, it is initiated by a triggering mechanism from the basic call process. The chain "link" pattern which describes the SF must then be obtained by the global service logic in order to process the SF. As new SFs are designed their SIB descriptions must be made available to the global service logic.

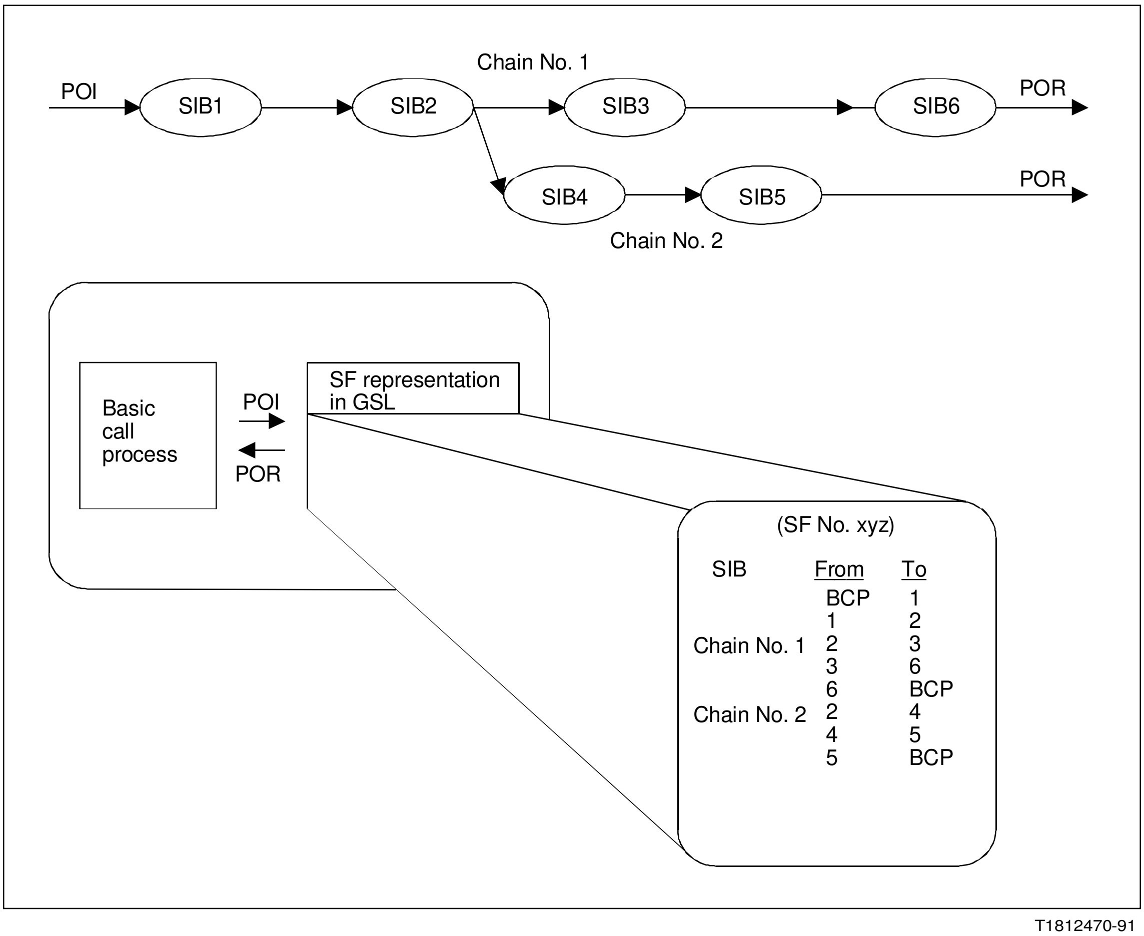


FIGURE 7

**Global functional plane blueprint of a service feature**

## ANNEX A

(to Recommendation I.329/Q.1203)

**Alphabetical list of abbreviations used in this Recommendation**

| | |
|---|---|
| BCP | Basic call process |
| CID | Call instance data |
| CIDFP | CID field pointer |
| CLI | Calling line identity |
| DFP | Distributed functional plane |
| FE | Functional entity |
| GFP | Global functional plane |
| GSL | Global service logic |

| | |
|---|---|
| IF | Information flow |
| IN | Intelligent network |
| INCM | IN conceptual model |
| PIN | Personal identification number |
| POI | Point of initiation |
| POR | Point of return |
| SCF | Service control function |
| SDF | Service data function |
| SDL | Specification and description language |
| SF | Service feature |
| SIB | Service independent building block |
| SSD | Service support data |
| SSF | Service switching function |
| UPT | Universal private telecommunication |
| VPN | Virtual private network |