INTERNATIONAL TELECOMMUNICATION UNION

CCITT

I.312/Q.1201

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

(10/92)

INTEGRATED SERVICES
DIGITAL NETWORK (ISDN)
I.312 (10/92)

OVERALL NETWORK ASPECTS
AND FUNCTIONS,
ISDN USER-NETWORK INTERFACES
Q.1201 (10/92)

PRINCIPLES OF INTELLIGENT NETWORK ARCHITECTURE

Recommendation I.312 / Q.1201

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.312/Q.1201 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.312 / Q.1201

PRINCIPLES OF INTELLIGENT NETWORK ARCHITECTURE

(1992)

1 Objectives, overall description

1.1 *Motivation, objectives, scope of Intelligent Network*

1.1.1 *Motivation*

The term Intelligent Network (IN) is used to describe an architectural concept which is intended to be applicable to all telecommunications networks. IN aims to ease the introduction of new services (i.e. Universal Personal Telecommunication (UPT), Virtual Private Network (VPN), Freephone, etc.) based on greater flexibility and new capabilities.

IN Recommendations are motivated by the interests of telecommunication services providers to rapidly, cost effectively and differentially satisfy their existing and potential market needs for services. Also, these service providers seek to improve the quality and reduce the cost of network service operations and management.

Additionally, current trends in technology permit a greater degree of intelligence and greater freedom in the allocation of intelligence in the telecommunications network. For example, the improved mobility derived from miniaturization of electronic components allows for a greater degree of distributed functionality within and between service provider networks. Factors permitting such intelligence include: advances in digital transmission and switching, common channel signalling, distributed data processing, data base management and expert systems.

1.1.2 *Objectives of Intelligent Network*

The objective of IN is to allow the inclusion of additional capabilities to facilitate provisioning of service, independent of the service/network implementation in a multi-vendor environment. Service implementation independence allows service providers to define their own services independent of service specific developments by equipment vendors.

Network implementation independence allows network operators to allocate functionality and resources within their networks and to efficiently manage their networks independent of network implementation specific developments by equipment vendors.

1.1.3 *Scope of Intelligent Network*

Types of networks: IN is applicable to a wide variety of networks, including but not limited to: public switched telephone network (PSTN) mobile, packet switched public data network (PSPDN) and integrated services digital network (ISDN) – both narrow-band-ISDN (N-ISDN) and broadband-ISDN (B-ISDN).

Type of services: IN supports a wide variety of services, including supplementary services, and utilizes existing and future bearer services (e.g. as those defined in N-ISDN and B-ISDN contexts).

1.2 *Definition of Intelligent Network*

Intelligent Network (IN) is an architectural concept for the operation and provision of new services which is characterized by:

– extensive use of information processing techniques;

– efficient use of network resources;

– modularization and reusability of network functions;

- integrated service creations and implementation by means of the modularized reusable network functions;
- flexible allocation of network functions to physical entities;
- portability of network functions among physical entities;
- standardized communication between network functions via service independent interfaces;
- service subscriber[1] control of some subscriber-specific service attributes;
- service user[2] control of some user-specific service attributes;
- standardized management of service logic.

1.3 *Evolution of Intelligent Network Recommendations*

A phased standardization process is recommended. This Recommendation takes into account the fact that the specification and the deployment of networks that meet all the objectives of the IN target architecture will take many years. In addition, IN as a new architectural concept should be introduced starting from the existing networks and the current Recommendations.

Furthermore, the IN target (long-term views) will evolve, reflecting operational experiences, new technological opportunities and market evolution.

In order to ensure smooth evolution towards the target, IN Recommendations shall allow:

- backward compatibility of each evolutionary phase;
- open-endedness towards long-term views.

In particular, backward compatibility implies utilization of previous Recommendations within a new phase.

1.3.1 *General considerations on the standardization process*

Figure 1 shows how the different aspects of the standardization process are related, distinguishing:

- IN concept and modelling, which are influenced by the current network Recommendations; and
- their application to long-term views and to Recommendations on intermediate phases (transitional IN Recommendations).

The influence of current network Recommendations and of long-term views on transitional IN Recommendations is also shown.

1.3.2 *Recommendation areas*

A Recommendations outline can be derived from the above described standardization process. Three areas are fundamental to the production of Recommendations:

1.3.2.1 *Area 1 – IN architectural concept and modelling*

This area contains the IN concept, modelling techniques and other network design tools, as well as the results from the development of the IN target architecture. This area is documented in the I.320-Series/Q.120Y-Series.

1.3.2.2 *Area 2 – IN transition planning and phase definitions*

Here guidelines are worked out allowing a proper transition from the existing technology base towards a target IN infrastructure. For each phase it is necessary to determine the service functionality (e.g. whether to include service creation capabilities for the customer) and technological constraints (e.g. using the D-channel for ISDN packet switched services). This area is documented in the Q.1200 Series Recommendations.

---

[1] A service subscriber (customer) is a person or entity who, or which, obtains a service from a service provider and is responsible for the payment of the charges due to that service provider.

[2] A service user is a person who has access to and makes use of services.

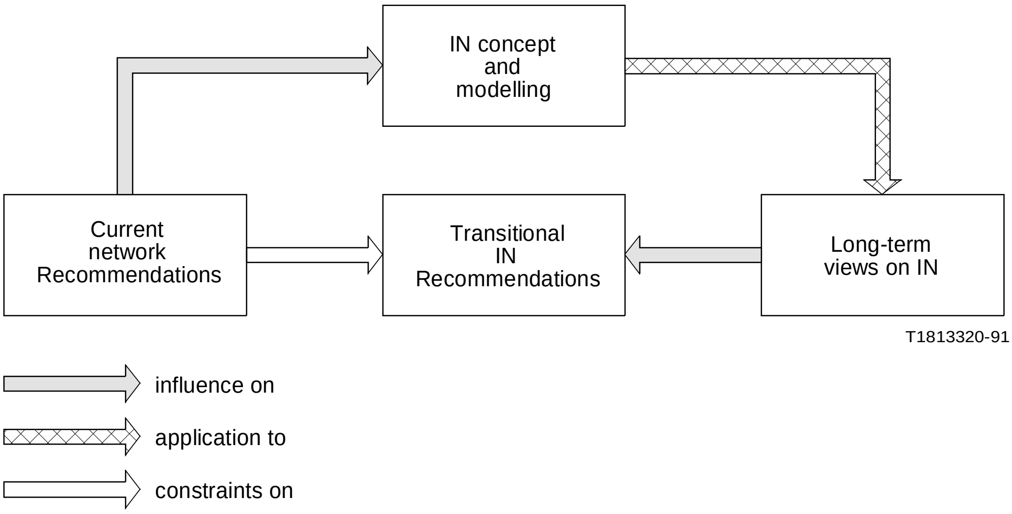


FIGURE 1

Relationship between the aspects of the standardization process

1.3.2.3 *Area 3 – IN architecture and interfaces for each phase*

In this area the specifications are provided that are necessary for the implementation of IN equipment, interfaces etc. For each phase an evolving set of Recommendations will be developed. This area is documented in the Q.12xy-Series ($1 \le x \le 9$, $2 \le y \le 9$).

The relationship between the areas is shown in Figure 2.

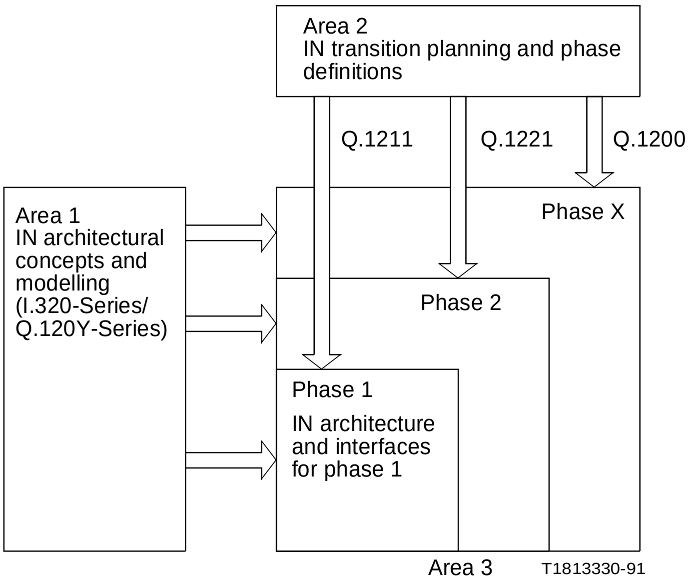


FIGURE 2

Recommendations areas

1.3.3 *Phased standardization and definition of capability sets*

Capability sets (CS) are defined as sets of IN capabilities which are to be subjects of standardization activities and for which the availability of Recommendations will be targeted for a particular evolution phase.

The long-term capability set (LTCS) is the CS for the target IN architecture.

The sequencing of CSs is shown in Figure 3. The figure also indicates the relationship between the previous defined areas and the definition of each capability set.

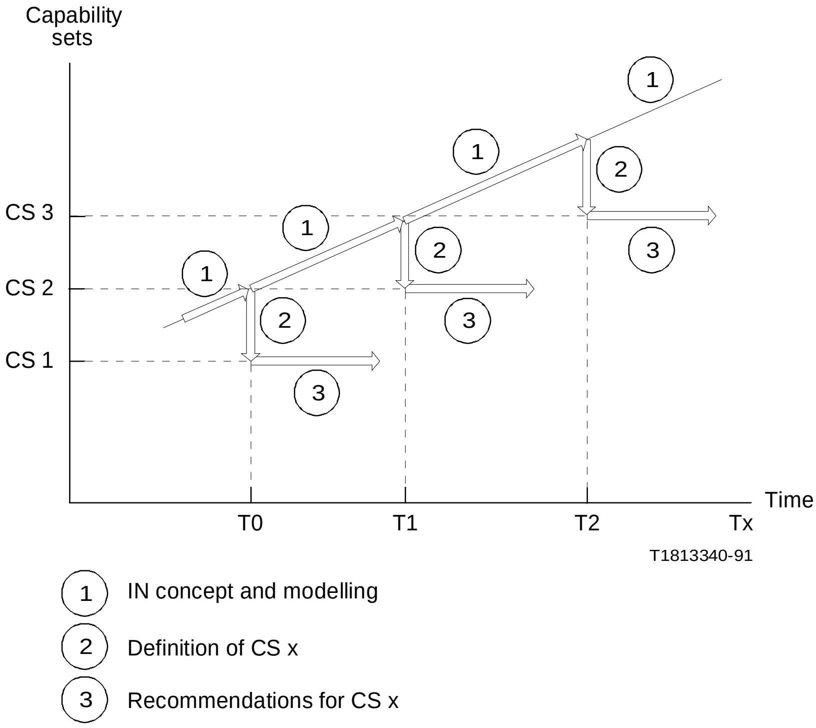


FIGURE 3
Sequencing of Capability sets (CSs)

2 IN functional requirements

2.0 *Introduction*

IN functional requirements arise as a result of the need to provide network capabilities for both

– customer needs (service requirements); and

– network operator needs (network requirements).

A service user is an entity external to the network that uses its services. A service is that which is offered by an Administration to its customers in order to satisfy a telecommunications requirement. Part of the service used by customers may be provided/managed by other customers of the network.

Service requirements will assist in identifying specific services that are offered to the customer. These service capabilities are also referred to as (telecommunication) services. Network requirements span the ability to create, deploy, operate and maintain network capabilities to provide services. The categorization of service requirements versus network requirements is schematically shown in Figure 4.

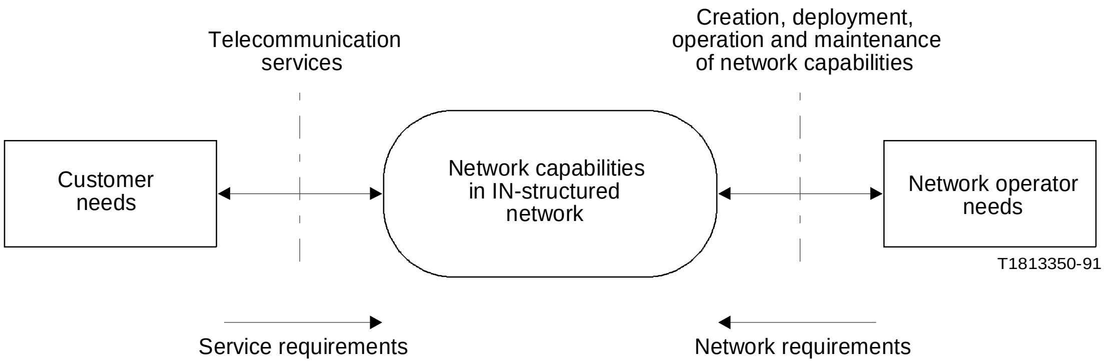


FIGURE 4

Service requirements versus network requirements

Service and network requirements can be identified for the following areas of service/network capabilities: service creation, service management, network management, service processing and network interworking.

– *Service creation:* An activity whereby supplementary services are brought into being through specification phase, development phase and verification phase.

– *Service management:* An activity to support the proper operation of a service and the administration of information relating to the user/customer and/or the network operator. Service management can support the following processes: service development, service provisioning, service control, billing and service monitoring.

– *Network management:* An activity to support the proper operation of an IN-structured network.

– *Service processing:* Consists of basic call and supplementary service processing which are the serial and/or parallel executions of network functions in a coordinated way, such that basic and supplementary services are provided to the customers.

– *Network interworking:* A process through which several networks (IN to IN or IN to non-IN) cooperate to provide a service.

Figure 5 gives a general overview of these capability areas including their relation to service and network requirements. The network interworking capabilities are not shown in this figure as these are indirectly contained in the other capability areas.

2.1 *Service requirements*

2.1.1 *Overall requirements*

The following requirements may also apply to existing networks. Nevertheless, they are stated here to underline their importance when defining the IN architecture:

– it should be possible to access services by the usual user network interface (e.g. POTS, ISDN);

– it should be possible to access services that span multiple networks;

– it should be possible to invoke a service on a call-by-call basis or for a period of time, in the latter case the service may be deactivated at the end of the period;

– it should be possible to perform some access control to a service;

– it should be easy to define and introduce services;

– it should be possible to support services involving calls between two or more parties;

– it should be possible to record service usage in the network (service supervision, tests, performance information, charging);

– it should be possible to provide services that imply the use of functions in several networks;

– it should be possible to control the interactions between different invocations of the same service.

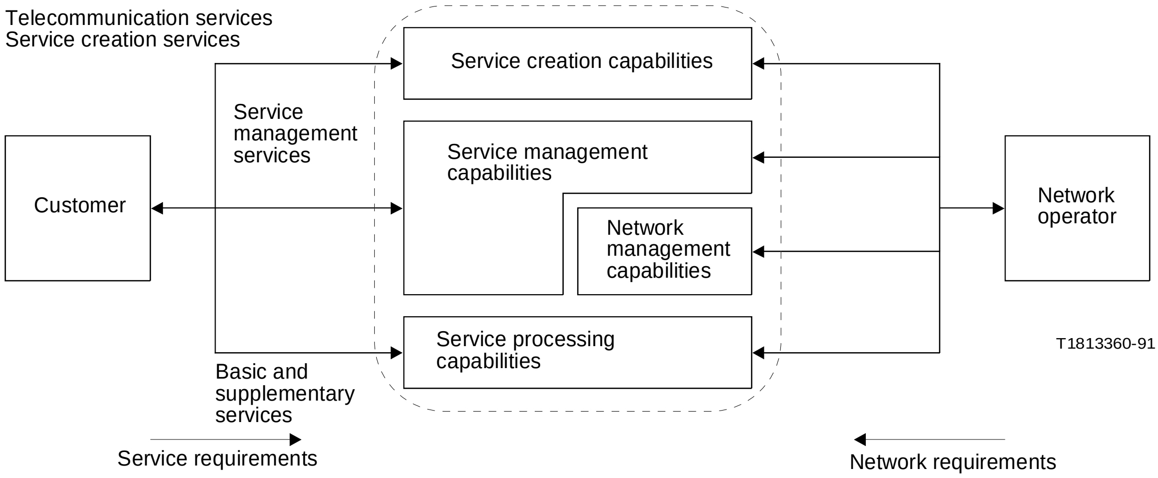


FIGURE 5
Network capabilities in IN-structured network

2.1.2 *Service creation*

A subset of the service creation capabilities used by the network operator (described in § 2.2.2) may be offered to customers. Service requirements for service creation refer to the network capabilities that are used by network operators for the provision of service creation services to customers. This is schematically shown in Figure 6.

2.1.3 *Service management*

Service requirements for service management refer to the network capabilities that are necessary for the provision, from a customer's point of view, of service management services to customers. This is schematically shown in Figure 7.

A subset of the service management capabilities used by the network operator may be offered to customers.

2.1.3.1 *Service management during deployment phase*

(For further study.)

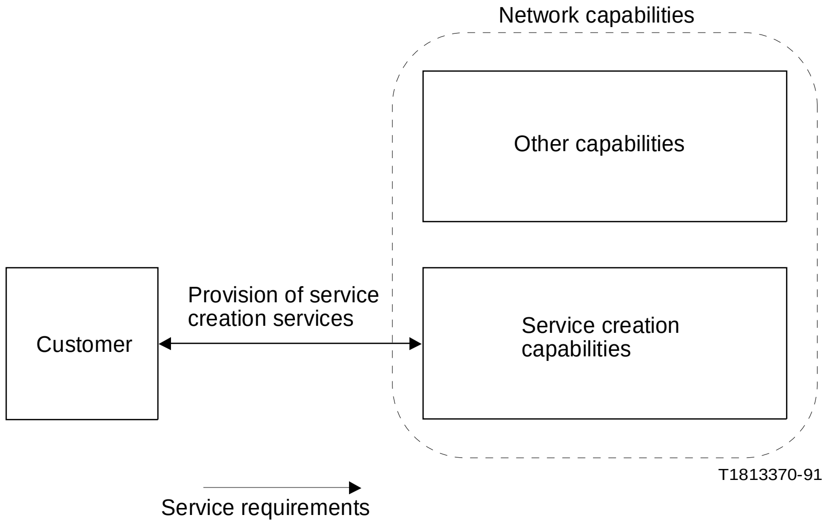


FIGURE 6
Service requirements for service creation

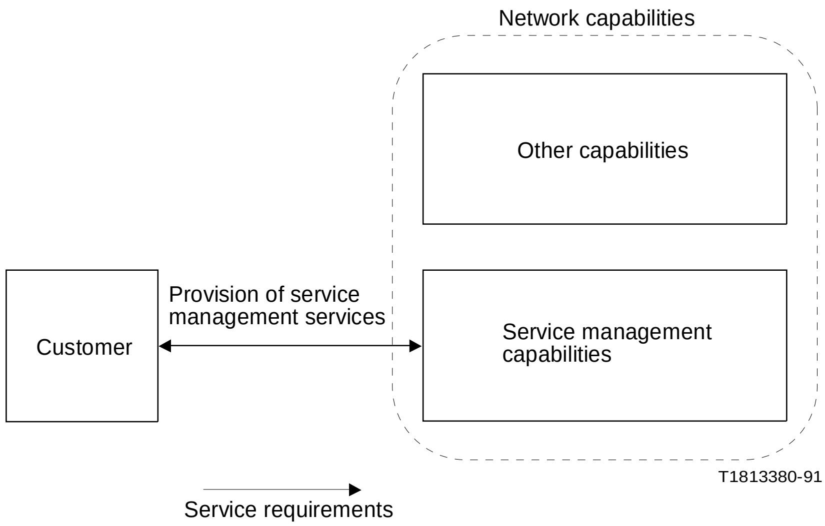


FIGURE 7
Service requirements for service management

2.1.3.2 *Service management during provisioning phase*

Service provisioning is the activity of installing and deploying the necessary functionality in appropriate network elements to realize a service to a specific customer along with the initial activation and customization. After provisioning the customer's service is administered.

2.1.3.3 *Service management during utilization phase*

This activity includes:

– activation, deactivation, service maintenance and service customization, after the service provisioning has taken place;

– service activation, which is the activity to make the service usable by a specific customer (e.g. call forwarding activation);

– service customization, which is the activity of setting up the appropriate service parameters to control the operation of service to meet the specific needs of the customer (e.g. setting Call Distribution percentages);

– charging, which is mainly to collect data on service usage and to generate reports thereon for billing either on demand or automatically. It includes the alteration of charge within the framework of agreement with a network operator. Other requirements include the preparation of customer-specific billing reports and data which are accessible to the customer;

– service monitoring, which provides the capability to collect and accumulate statistics on a given service with a view to determine the Quality of Service operation and adjust the operation to suit the prevailing conditions; also, the data may be used in the process of service creation to determine if an implementation of an IN supported supplementary service meets the service's performance requirements.

2.1.4 *Service processing*

Service requirements for service processing refer to the network capabilities that are necessary for the provision, from a customer's point of view, of basic and supplementary services by an IN-structured network. This is schematically shown in Figure 8.

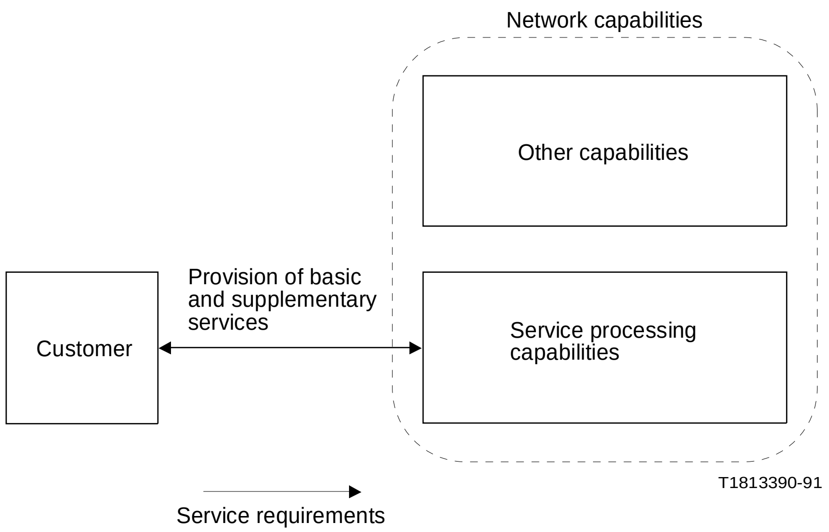


FIGURE 8
Service requirements for service processing

The IN is primarily a network concept that aims for efficient creation, deployment and management of supplementary services that enhance basic services. With regard to the provision of basic and supplementary services, the IN concept is "transparent" to the customer, i.e. the customer is unaware of whether a service is provided in an IN way. This "transparency" basically implies that, from a customer's point of view, no service processing requirements can be identified that have specific reference to the IN as such. Notwithstanding this, the IN should be capable of supporting a broad range of basic and supplementary services. Service processing requirements can be identified for the following capabilities:

– service capabilities that are necessary to support a broad range of basic and supplementary services;

– access capabilities that are necessary to interface with the network and to access the services.

The relationship between service capabilities and access capabilities is visualized in Figure 9.

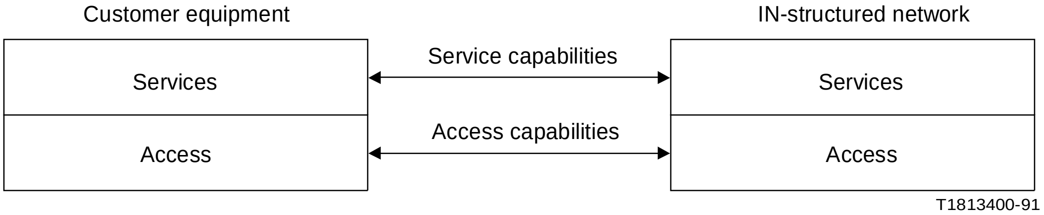


FIGURE 9

Service capabilities and access capabilities

2.1.4.1 *Service capabilities*

Requirements on service capabilities directly result from the services that are to be supported by the network. Service capabilities are related to basic services (bearer and tele-) and, in particular, to the supplementary services that may enhance these basic services. As stated above, the IN is primarily a network concept for the support (creation, deployment, etc.) of supplementary services. Because supplementary services can only be provided in combination with basic services, it is necessary to define for which basic services the IN concept can be applied. In this respect the IN concept can be applied to the support of supplementary services for the following basic services:

a) *bearer services*:

  – circuit-mode unrestricted (various bit rates);

  – circuit-mode speech;

  – circuit-mode audio;

  – packet switched data services;

  – circuit switched data services;

  – others;

b) *teleservices*:

  – telephony;

  – telefax;

  – videotex;

c) *broadband interactive services*:

  – conversational service;

  – messaging services;

  – retrieval services;

d) *broadband distribution services*:

– distribution services without user individual presentation control;

– distribution services with user individual presentation control;

– others;

With regard to the provision of supplementary services, the network should support service-independent network functions in a way that will facilitate rapid service introduction.

2.1.4.2 *Access capabilities*

In order to use a particular service, the customer needs an access arrangement to the network(s), that is providing the service. Nevertheless, the customer is mainly interested in the services themselves and not in the specific access arrangement that is used to physically connect to the network. For example, it should be possible that a single Virtual Private Network service can be accessed by various access arrangements such as ISDN interfaces, POTS interfaces, "mobile" interfaces, etc. In this case the VPN service could span multiple networks with different interface technologies. Access arrangements via (non-IN) sub-networks should also be supported (see Figure 10). In case of multiple access arrangements to a service, it should be noted that a particular access arrangement may impose technical, operational and/or regulatory limitations on the provided service.

When a particular service spans multiple networks, network interworking arrangements will be required between the different networks to allow ubiquitous provision of the service. Requirements for network interworking capabilities are identified in § 2.2.6.

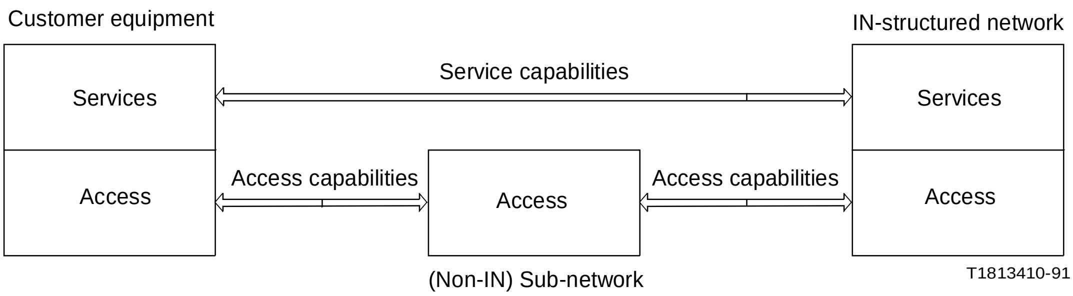


FIGURE 10
Access to services via (non-IN) sub-networks

In order to allow freedom of access to services, it is necessary that there be a sufficient degree of independence between the service capabilities on the one hand and the access capabilities on the other hand. The following access capabilities are foreseen for the IN:

a) fixed network:

– PSTN access;

– ISDN access;

– PSPDN access;

b) private network access

c) mobile network;

– PLMN access;

d) broadband network:

– asynchronous transfer mode (ATM) access;

– synchronous transfer mode (STM) access.

2.1.5 *Service interworking*

Service interworking describes special aspects of the individual service, if the service is used in a connection which exists partly inside a given IN-structured network and partly inside a non-IN structured network, or which, for certain operational aspects, routes through more than one IN-structured network.

2.2 *Network requirements*

2.2.1 *Overall requirements*

The following reflects overall network requirements. Specific detailed requirements can be found in the relevant subsection:

- it should be possible to move cost-effectively from existing network bases to target network bases in a practical and flexible manner;
- it should be possible to reduce redundancies among network functions in physical entities;
- it should be possible to allow for the flexible allocation of network functions to physical entities;
- there is a need for communication protocols that allow flexibility in the allocation of functions;
- it should be possible to create new services from network functions in a cost and time efficient manner;
- it should be possible to guarantee the integrity of the network when a new service is being introduced;
- it should be possible to manage network elements and network resources such that quality of service and network performance can be guaranteed.

2.2.2 *Service creation*

Network requirements for service creation refer to the network capabilities that are necessary, from a network operator point of view, for the creation of new supplementary services. This is schematically shown in Figure 11.

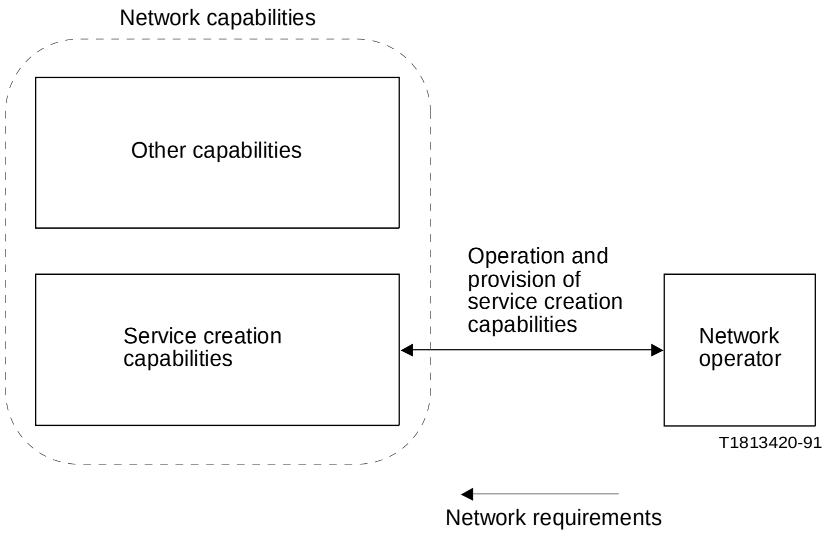


FIGURE 11

Network requirements for service creation

The creation of (new) services consists of several steps which are summarized in the service creation process. The different steps are:

– service specification;

– service development;

– service verification.

Service specification is the first step in the service creation process. As such, this step includes such activities as refinement of detailed service description requirements, functional analysis, generation and verification of a service specification and definition of a high level structured design. The primary outputs of this step include a service specification and a high level structured design which together provide sufficient detail to drive subsequent service development and verification steps.

Service development is the step which transforms a high level structured design into a detailed structured software design and subsequently develops the necessary software components, data definitions, etc. required to realize that design. The major output of this step is the developed service software and documentation which is ready for more rigourous service verification testing.

Service verification is the step in the service creation process where the developed service software (including supporting documentation) is rigourously tested to validate that the resulting service application completely satisfies the specification. The principal output of this step is thus the verified service software and supporting documentations required for deployment.

The service creation environment provides an environment which allows the easy creation of (new) services in a network-configuration and network-type independent way by means of service independent building blocks.

The service creation environment should, in an efficient and effective way, provide service-implementation independent tools, techniques, languages (e.g. specification languages) and procedures to support the service creation process in which service logic can be created. Network operators may choose to offer a subset of the service creation capabilities to service users.

There is a need to standardize a representation of service logic with standardized function calls (application programming interface) to IN capabilities.

In order to guarantee the integrity and security of the networks as well as to guarantee the integrity of each created service, it should be necessary to define the scope of accessible IN capabilities through the service creation activity. This scope isolates the unnecessary and unsolicited interactions between service and accesses to the network capability.

Each service can be represented by several different types of service logic. Service logic may be classified according to certain characteristics, based on its type of usage. Service logic must be implemented in a service-, network configuration-, and network-type (e.g. PSTN, ISDN, PLMN) independent way. This can best be done by means of service independent building blocks. These building blocks are combined by means of sequential and conditional programming statements using selected programming languages.

Since service logic may be classified according to its usage, building blocks (related to the service logic) can be similarly classified.

Some examples of the usage of service logic are:

– service logic for the execution aspects of a service (e.g. service processing logic, statistics logic, data base logic, charging logic);

– service logic for the management aspects of a service (e.g. statistics logic, charging logic, data base logic, service-data related logic, user-data related logic).

2.2.3 *Service management*

Network requirements for service management refer to the network capabilities that are necessary, from a network operator point of view, to support the proper operation of services. This is schematically shown in Figure 12.

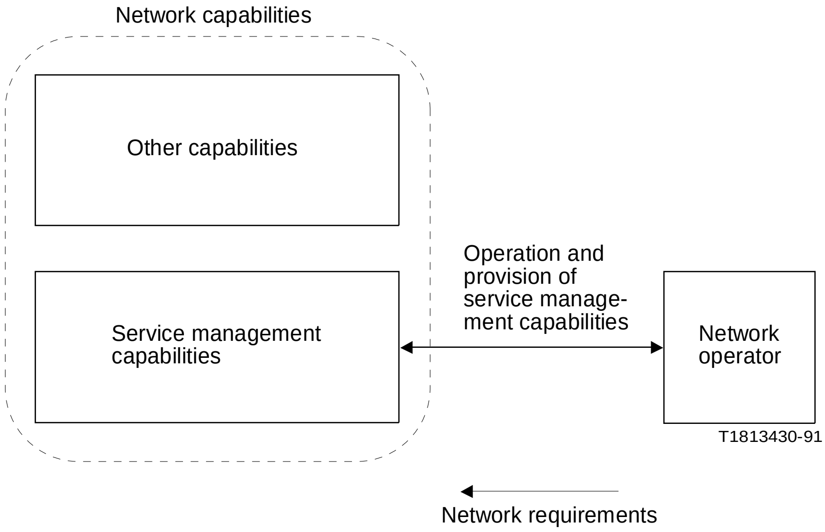


FIGURE 12

Network requirements for service management

2.2.3.1 *Service management during deployment phase*

This activity requires first a limited deployment of service for testing purposes. If testing is successful follow with:

– downloading of service logic, management logic, etc., into appropriate network elements, and

– activation of logic.

2.2.3.2 *Service management during provisioning phase*

This involves such activities as the creation of the customer’s service profile in the service logic and the activation of the service for the customer. This may require the initialization of trigger conditions and the creation of the customer’s service profile in the service logic.

2.2.3.3 *Service management during utilization phase*

– Service control gives the customers the ability to modify parameters which control their service. This could include, for example, screening lists or rules for routing calls by time of day and/or day of week.

– Data entered by the customer must be validated to insure that, for example, the customer has the permission to change the indicated parameters and that the parameters, as modified, are valid.

– Under IN, there may need to be several kinds of charging schemes, e.g. flat rate and usage sensitive. The usage sensitive charging may consider not only the duration of a call, but also the network resources required to provide the service. In this latter case, the tariff may be “service independent” in the sense that it is computed from the cost of the network resources used rather than from the value to the customer of the service offered.

Service monitoring can be either automatic or manual. Typical manual operations include queries regarding status information and network configuration.

Data gathered automatically could include traffic loads and other performance data. The following activities support automatic service monitoring:

- creation and modification of measurement schedules which include such fields as the measurements to be made, the frequency of data collection for each measurement, and the format and frequency of each report;
- collection and validation of data requested by the schedules;
- formatting and delivering reports as requested.

2.2.4 *Network management*

Network requirements for network management refer to the network capabilities that are necessary, from a network operator point of view, to support the proper operation of the IN-structured network. This is schematically shown in Figure 13.

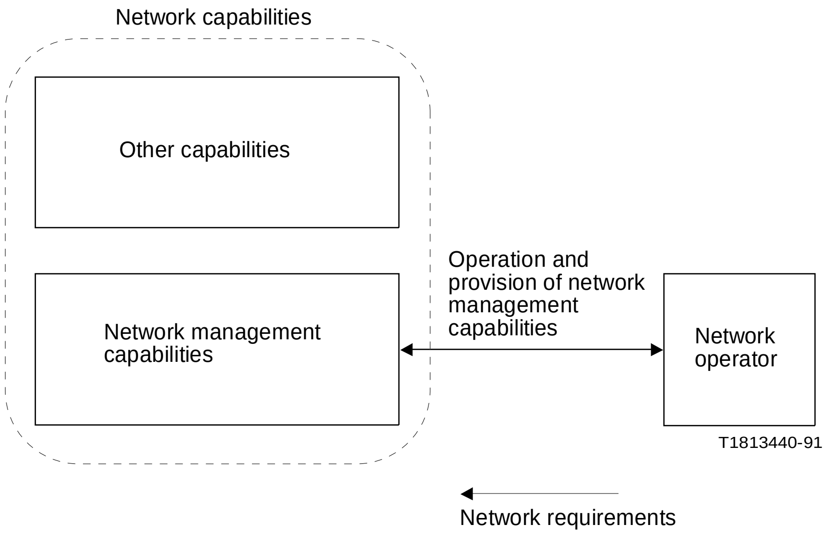


FIGURE 13

Network requirements for network management

The IN and non-IN requirements of network management are not essentially very different. The TMN application functions are relevant to IN as much as they are to non-IN. Therefore, the following organization of management capabilities (see Recommendation M.30) are applicable:

1) performance management:
   - performance monitoring;
   - traffic management and network management;
   - quality of Service observations;
2) fault (maintenance) management:
   - alarm surveillance;
   - failure localization;
   - testing;

3) configuration management:

   – provisioning;

   – status and control;

   – installation;

4) accounting management;

5) security management.

Additional application functions may be needed to deal with the IN situation.

2.2.5 *Service processing*

Network requirements for service processing refer to the network capabilities that are necessary for the provision, from a network operator point of view, of basic and supplementary services by an IN-structured network. This is schematically shown in Figure 14.

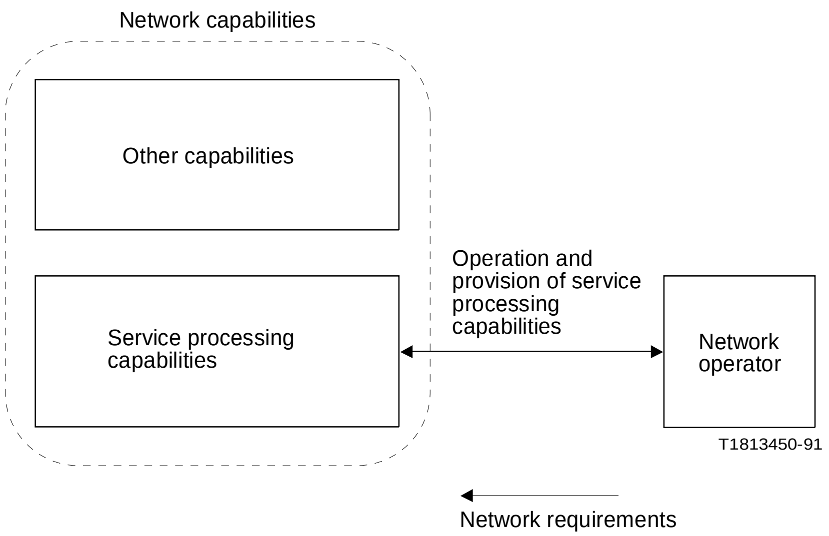


FIGURE 14

Network requirements for service processing

The main network requirements for service processing stem from the inability of network operators of traditional "non-IN" networks to rapidly create and deploy new supplementary services. To overcome this inability the IN aims for:

- rapid service implementations by means of reusable network functions;
- modularization of network functions;
- standardized communication between network functions via service independent interfaces.

To better understand how the IN can achieve the goal of fast service implementation, an overall comparison is made between traditional non-IN service processing (not meeting the aims listed above) and newly required IN service processing (meeting the aims listed above). Both cases are respectively referred to as the non-IN service processing model and the IN service processing model.

2.2.5.1 *Non-IN service processing model*

A simplified but typical representation of a non-IN service processing model is illustrated in Figure 15.

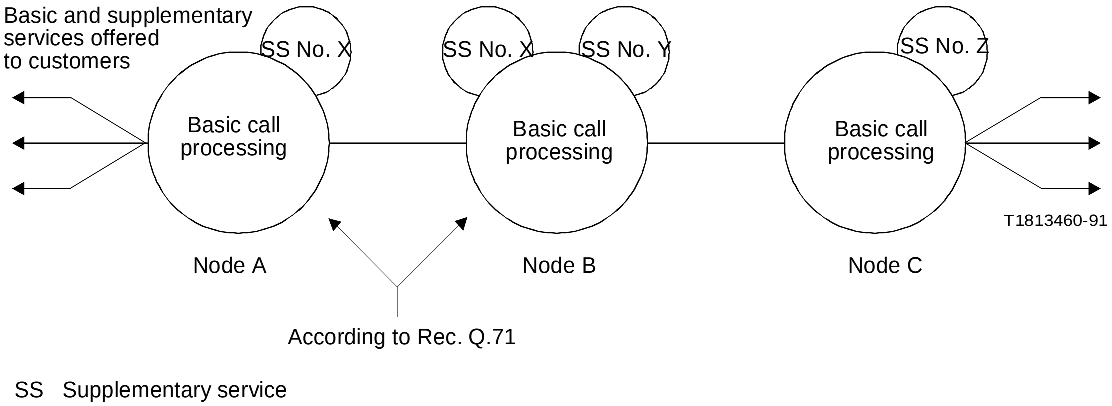


FIGURE 15

Non-IN service processing model

2.2.5.2 *IN-service processing model*

A high level overview of a desirable IN service processing model is illustrated in Figure 16.

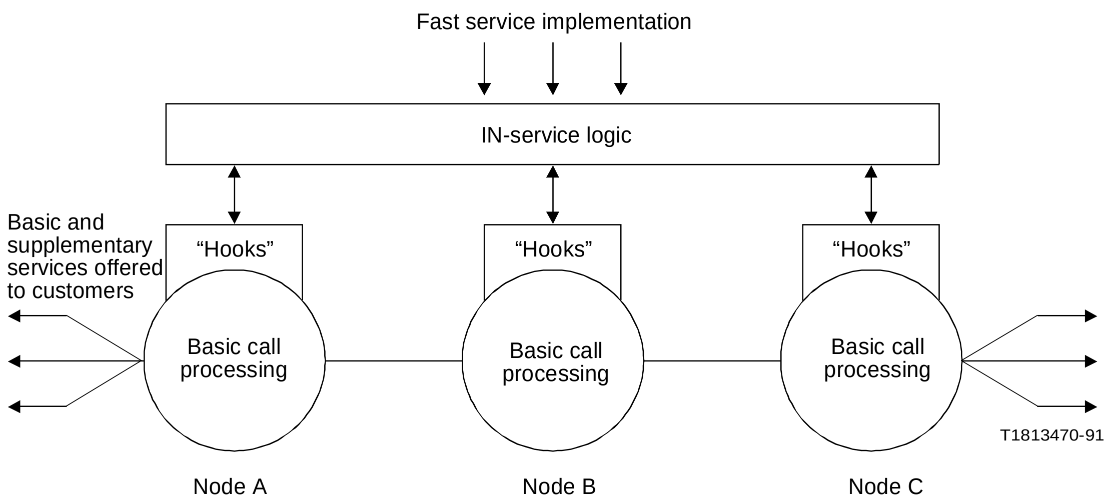


FIGURE 16

IN service processing model

The three main elements of this model are: the basic call processes, the "hooks" that allow the basic call processes to interact with IN service logic, and IN service logic that can be "programmed" to implement new supplementary services. For these elements the main principles are described below:

– The basic call process should be available all over the network and is designed to support, with optimal performance, services that do not require special features. In order to achieve flexibility in service processing, the basic call process needs to be modularized into service-independent sub-processes such that these can be executed autonomously (without interference from the outside during execution).

– "Hooks" are to be added to the basic call process forming the links between the individual basic call sub-processes and the service logic. The "hooks" are able to start an interaction session with the IN service logic. For this it should continuously check the basic call process for the occurrence of conditions on which an interaction session with IN service logic should be started. During an interaction session the basic call process can be temporarily suspended.

– IN service logic uses a programmable software environment that needs to be developed to allow fast implementation of new supplementary services. New supplementary services can be created by means of "programs" containing IN service logic. The IN service logic is able, via the "hooks" functionality, to interact with the basic call process. In this way IN service logic can control the sub-processes in the basic call process and the sequencing of these sub-processes.

  Thus, by changing logic at the service control point and modifying network data, a new service that uses existing network capabilities can readily be implemented.

  In addition IN service logic can decide to terminate an interaction session with the basic call process. The basic call process will then resume its execution as specified by the IN service logic. In order to allow fast service implementation, the IN service logic should have a logical view of the network resources that constitute the basic call process and additional (specialized) network functions.

– For proper service processing, the following principles apply:

  – it should be possible to distribute resources between services in a well balanced way;

  – it should be possible for IN supported services to share resources with non-IN supported services;

  – it should be possible to provide a different method of resource data management from the current embedded method;

  – it should be possible to introduce IN supported services specific resources.

2.2.5.3 *Consequences for call modelling in an IN architecture*

As part of the activities to define an IN architecture including the network elements within this architecture, there is a need for a call model that describes the real-time behaviour of call control capabilities for the provision of basic and supplementary services. In order to be consistent with the principles of the above-described IN service processing model, the IN call model should cover the following aspects:

– it should specify which basic services can be supported by the model;

– it should model the basic call processes (each individual basic service may require its own IN basic call process);

– it should describe trigger mechanisms ("hooks") that allow the IN basic call process to interact with service logic;

– it should provide a logical view (from the service logic point of view) of call processing functions and network resources, which as a consequence allows fast service implementation;

– it should specify the mechanisms according to which an IN-basic call process may interact with the service logic (e.g. single-ended interactions, simultaneous interactions, service-logic initiated interactions, etc.);

– it should be evolvable from the existing technology base.

2.2.6 *Network interworking*

Network interworking is a process in which several networks (IN to IN or IN to non-IN) cooperate to provide a service.

The need for network interworking capabilities results from the fact that customers may want to access services which span multiple networks. These networks may have different access types (PSTN, ISDN, etc.) and may have different levels of IN structuring (full, partial or no IN structuring). Irrespective of the access type and of the level of IN structuring, services should be provided to customers in a consistent way.

Network interworking requirements exist at different levels:

– service processing;

– service management;

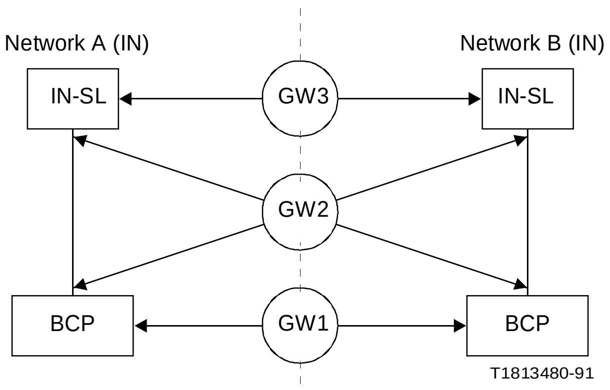
– service creation,

and in each level, some network interworking related gateway functions need to be defined.

2.2.6.1 *Gateway functions for service processing*

Figure 17 represents possible "gateway" functions which support network interworking at the level of service processing, when two IN-structured networks cooperate to provide a service.

A gateway function may be used, for example, to access service logic in other networks (GW2), or to provide communication between pieces of service logic pertaining to different networks (GW3). GW1 is used to route the call between the networks.



IN-SL IN-service logic
BCP Basic call process
GW Gateway process

Note 1 – It is anticipated that substantial service capabilities will result from earlier implementation of GW1 and GW3.

Note 2 – GW2 requires careful further study, in particular from the network security and integrity point of view.

FIGURE 17

Possible gateway functions between two INs

Figure 18 represents the use of network interworking between an IN-structured and a non-IN structured network.

A gateway function (GW4) is required to route the call between the IN-structured and non-IN-structured networks and to provide interworking between the BCP of network A and the BCP of network B.

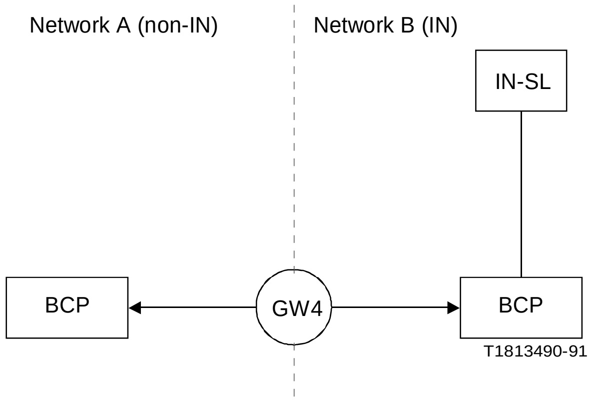


FIGURE 18

Gateway function between IN and non-IN

2.2.6.2 *Gateway functions for service management*

Figure 19 represents a gateway function for service management. At this level, a gateway function is required to link the service management process of the different networks which interwork. The gateway function should support service management during deployment, provisioning and utilization phases for services that span multiple networks.

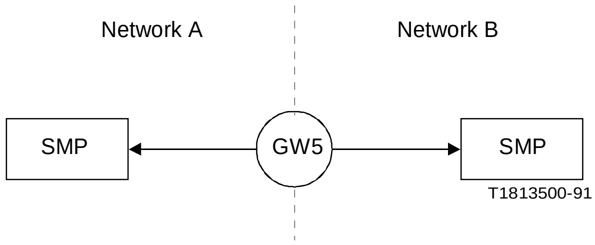


FIGURE 19

Gateway function for service management

2.2.6.3 *Gateway functions for service creation*

Service creation is an off-line activity and normally exists in a single network. The need for network interworking at the service creation level is for further study.

3 IN architectural concept

A key objective of the IN is to provide service-independent functions that can be used as "building blocks" to construct a variety of services. This allows easy specification and design of new services.

A second key objective is network-implementation-independent provision of services. This objective aims to isolate the services from the way the service-independent functions are actually implemented in various physical networks, thus providing services that are independent of underlying physical network infrastructure(s).

The network implementation independence has these objectives:

– services can use distributed network functions in various ways;

– services can span several networks and are independent of specific implementations in these networks;

– services can be independent of technological developments and evolution in network infrastructure, so that physical networks can evolve without affecting existing services;

– physical elements in such a network can be procured from different vendors.

3.1 *IN Conceptual Model (INCM)*

The IN Conceptual Model should not be considered in itself an architecture. It is a framework for the design and description of the IN architecture, taking into account guidelines in § 1, and in particular, the evolution of IN with its various phases.

Various "models" and "concepts" will be used in the standardization of IN. The INCM is intended to represent an integrated, formal framework within which these concepts are identified, characterized and related. It should be possible to define clearly the purpose, value and limitation of any IN concept and its relationship to other such concepts. Existing concepts may need to be adapted for use within this framework. To achieve this, the INCM consists of four "planes" where each plane represents a different abstract view of the capabilities provided by an IN-structured network. (See Figure 20). These views address service aspects, global functionality, distributed functionality and physical aspects of an IN.

3.1.1 *Service plane*

The service plane represents an exclusively service-oriented view. This view contains no information whatsoever regarding the implementation of the services in the network, e.g. an "IN-type" implementation is not visible. All that is perceived is the network's service-related behaviour as seen, for example, by a service user. Services are composed of one or more Service Features (SFs), which are the "lowest level" of services.

3.1.2 *Global functional plane*

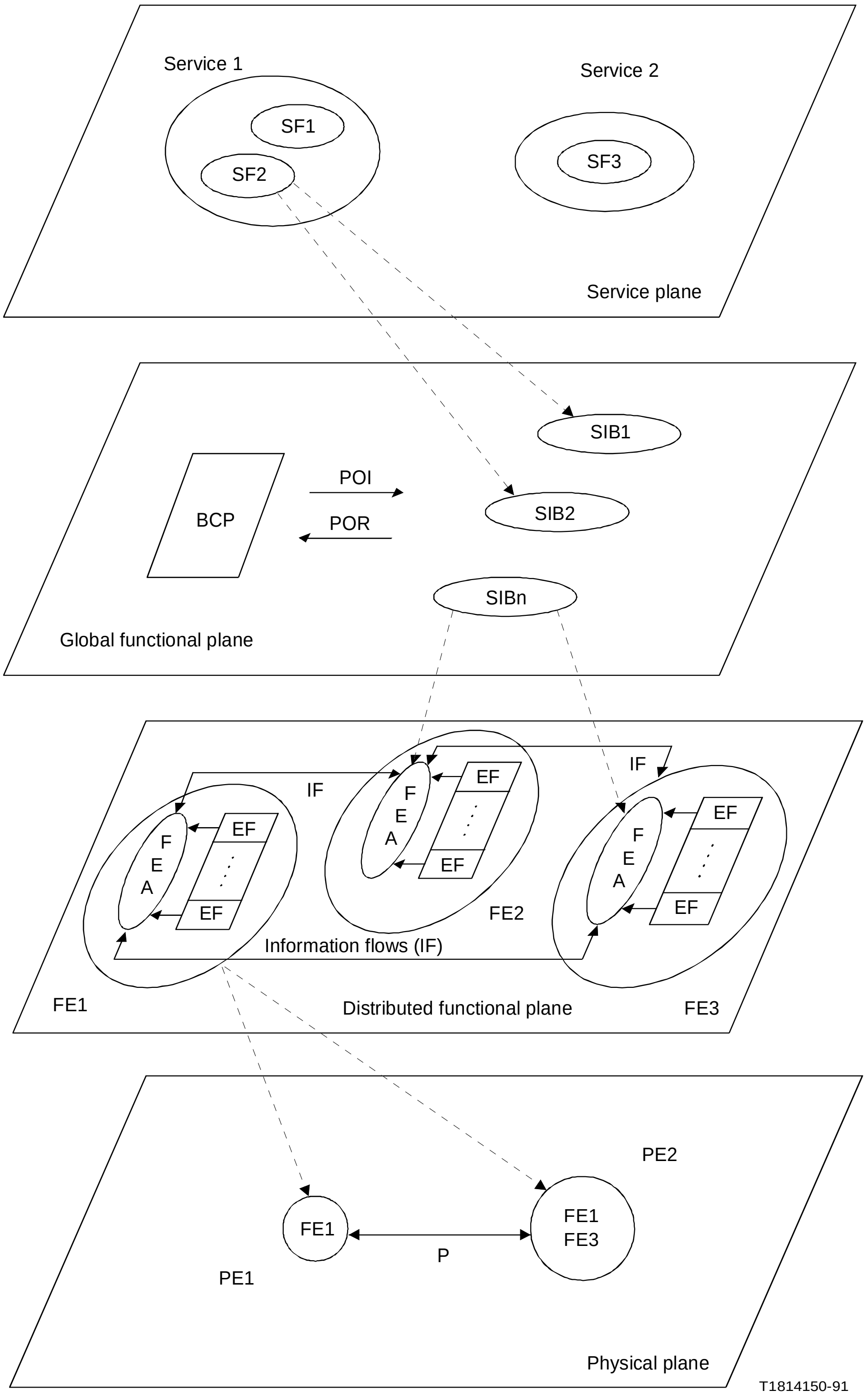
The global functional plane (GFP) models an IN-structured network as a single entity. Contained in this view is a global (network-wide) basic call processing (BCP) SIB, the service independent building blocks (SIBs), and point of initiation (POI) and point of return (POR) between the BCP and a chain of SIBs.

3.1.3 *Distributed functional plane*

The distributed functional plane (DFP) models a distributed view of an IN-structured network. Each functional entity (FE) may perform a variety of functional entity actions (FEAs). Any given FEA may be performed within different functional entities. However, a given FEA may not be distributed across functional entities.

Within each functional entity, various FEAs may be performed by one or more elementary functions. The manner in which elementary functions result in FEAs is for further study.

Service-independent building blocks (SIBs) are realized in the distributed functional plane (DFP) by a sequence of particular FEAs performed in the functional entities. Some of these FEAs result in information flows between functional entities.



BCP	Basic call process	PE	Physical entity
EF	Elementary function	POI	Point of initiation
FE	Functional entity	POR	Point of return
FEA	Functional entity action	SF	Service feature
IF	Information flow	SIB	Service-independent building block
P	Protocol	– – ►	Pointer

FIGURE 20

IN-Conceptual Model

3.1.4 *Physical plane*

The Physical Plane models the physical aspects of IN-structured networks. The model identifies the different physical entities and protocols that may exist in real IN-structured networks. It also indicates which functional entities are implemented in which physical entities.

3.1.5 *Relationship with the 3-stage method*

The Recommendation I.130 based 3-stage method needs enchancements for IN. The correspondence between the four planes of the IN Conceptual Model and the 3-stage method is as follows:

- Stage 1 methodology may be used to define services and services features in the service plane and to define SIBs in the global functional plane;
- Stage 2 description methodology may be used to define the realization of SIBs in the distributed functional plane;
- Protocols defined using stage 3 methodology may be applied in the physical plane.

3.1.6 *Service logic*

Service logic may have different representations within each plane (see Figure 21), e.g.:

- *Global functional plane:* There is one set of global service logic (GSL) per service feature and it uses SIBs.
- *Distributed functional plane:* There is one set of distributed service logic (DSL) per SIB and it uses FEAs and information flows.
- *Physical plane:* Service logic programs may be installed into and executed by any physical entity that contains that SCF functional entity.

3.1.7 **Application programming interface (API)**

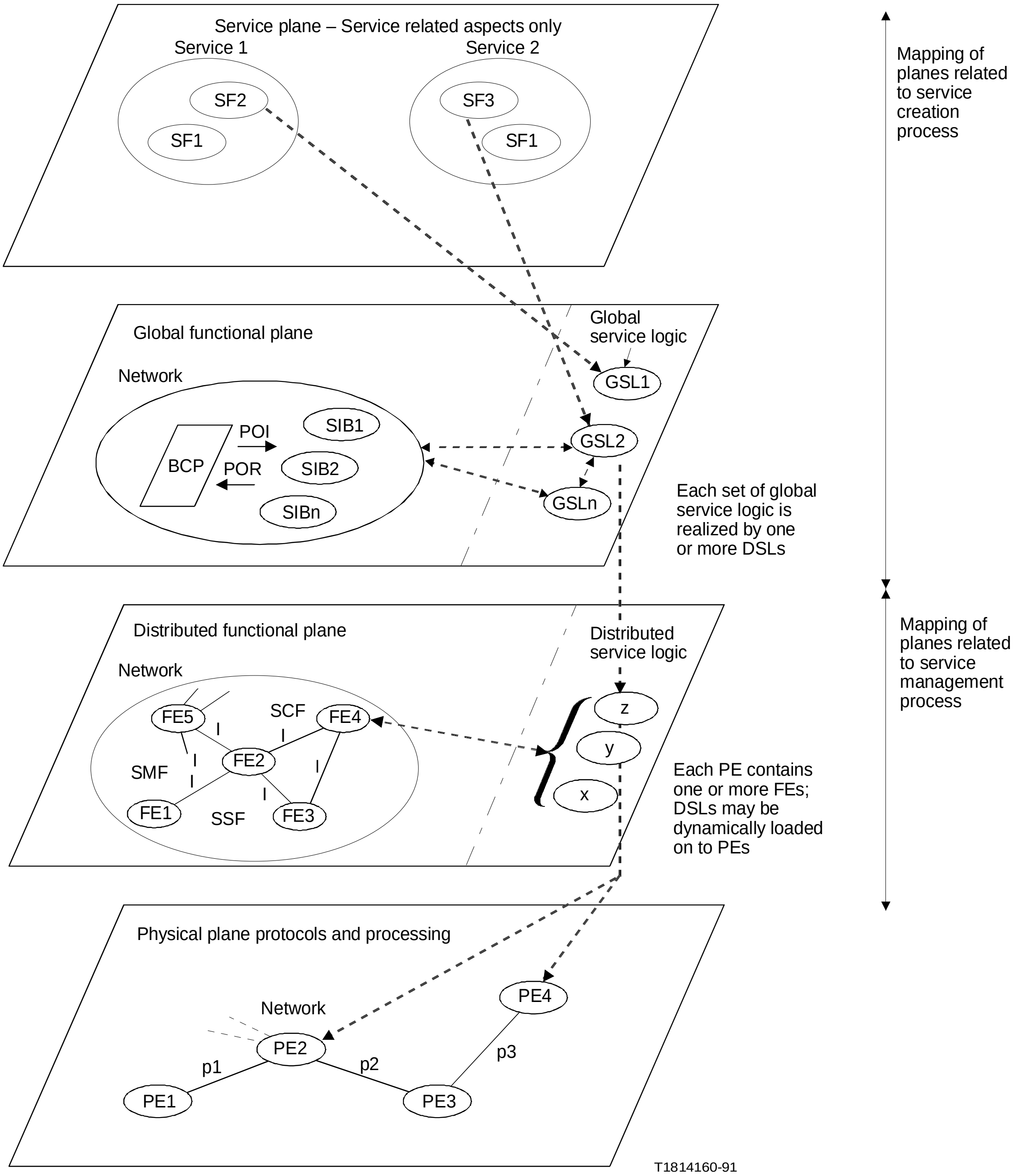
3.1.7.1 *General API definition*

An API provides a set of interfaces from an application environment to an execution environment. The execution environment provides services to the application environment.

3.1.8 *Relationships among different planes*

As noted in § 3.1, the entities contained in adjacent planes of the INCM are related to each other. The nature of the relationship is as follows:

- *Service plane to GF plane:* Service features within the service plane are realized in the GF plane by a combination of global service logic and SIBs including the basic call process SIBs. This mapping is related to the service creation process.
- *GF plane to distributed functional (DF) plane:* Each SIB identified in the GF plane must be present in at least one FE in the DF plane. A SIB may be realized in more than one FE. Thus, cooperation of several FEs may be needed. The service logic in the GF plane maps onto one or more DSLs in the DF plane. This mapping is related to the service creation process.
- *DF plane to physical plane:* FEs identified in the DF plane determine the behaviour of the physical entities (PEs) onto which they are mapped. Each FE must be mapped onto one physical entity, but, each PE contains one or more FEs. Relationships between FEs, identified in the DF plane, are specified as protocols in the physical plane. DSLs may be dynamically loaded into physical entities and this mapping is related to the service management process.



BCP	Basic call processing	POR	Point of return
DSL	Distributed service logic	SCF	Service control function
FE	Functional entity	SF	Service feature
GSL	Global service logic	SIB	Service independent building block
I	Interface	SMF	Service management function
p1, p2, p3	Protocols	SSF	Service switching function
PE	Physical entity	– – –	Service creation and management processes
POI	Point of initiation		

FIGURE 21

IN Conceptual Model with service logic
(The service creation and management processes are recognized as mapping of adjacent planes)

An example of these relationships is shown in Figure 22. Here it is shown that three services (Freephone, Virtual Private Network (VPN) and Universal Personal Telecommunication (UPT) can each potentially share various service features (A, B, C, etc.). These service features may be realized in the global functional plane by one or more SIBs, for example Screen and Compare.

These SIBs are realized in the distributed functional plane by functional entity actions (FEAs) (e.g. 1, 2, 3, 4) which occur within one or more FEs, e.g. service switching function (SSF), service control function (SCF) and service data function (SDF). When more than one FE is required, information flows result between FEAs as shown.

3.1.9 *Service interaction*

A set of rules must be provided at the service plane level; these services may be IN or non-IN. These interactions will also impact all the other planes. In addition to the set of rules at the service plane level, a robust mechanism for resolving easy feature interaction should be provided in other planes in a service-independent manner to facilitate rapid feature introduction.

3.1.10 *Service and network interworking*

3.1.10.1 *Service interworking*

(for further study)

3.1.10.2 *Network interworking*

1) *Network interworking in the distributed functional plane:*

   – this plane should be explicitly divided into several parts, each of which represents one functional network;

   – network interworking requires that relationships are defined between pairs of functional entities in different functional networks (e.g. between an SCF in network A and an SDF in network B);

   – each network interworking interaction between communicating pairs of functional entities is termed an information flow. The network interworking relationship between any pair of functional entities is the set of network interworking related information flows between them;

   – the semantic meaning and information content of each information flow needs to consider network interworking capabilities, network security and network integrity;

   – functional entities which support internetworking provide internetworking functionality based on the associated information flows and functional entity actions.

2) *Network interworking in the physical plane:*

   – this plane should be explicitly divided into several parts, each of which represents one physical network;

   – functional entities are allocated to physical entities in each of the networks that interwork;

   – network interworking protocols are defined and standardized to support the network interworking relationship between two functional entities, each of which is located in a different physical network.

3.1.11 *Management functionality*

Management is related to all planes of the IN Conceptual Model. In an IN-structured network there is a need to consider both the service and network aspects of management. Specific text on these aspects is contained within each architectural section (§§ 4 to 7).

One particular aspect of management (e.g. service creation, service introduction, service tailoring, customer control, etc.) can be viewed as non-real time, and independent of the actual real-time service execution. In a multivendor environment, different versions of physical representations of the same functional entity may exist, where versions may contain subsets of capabilities of other versions.

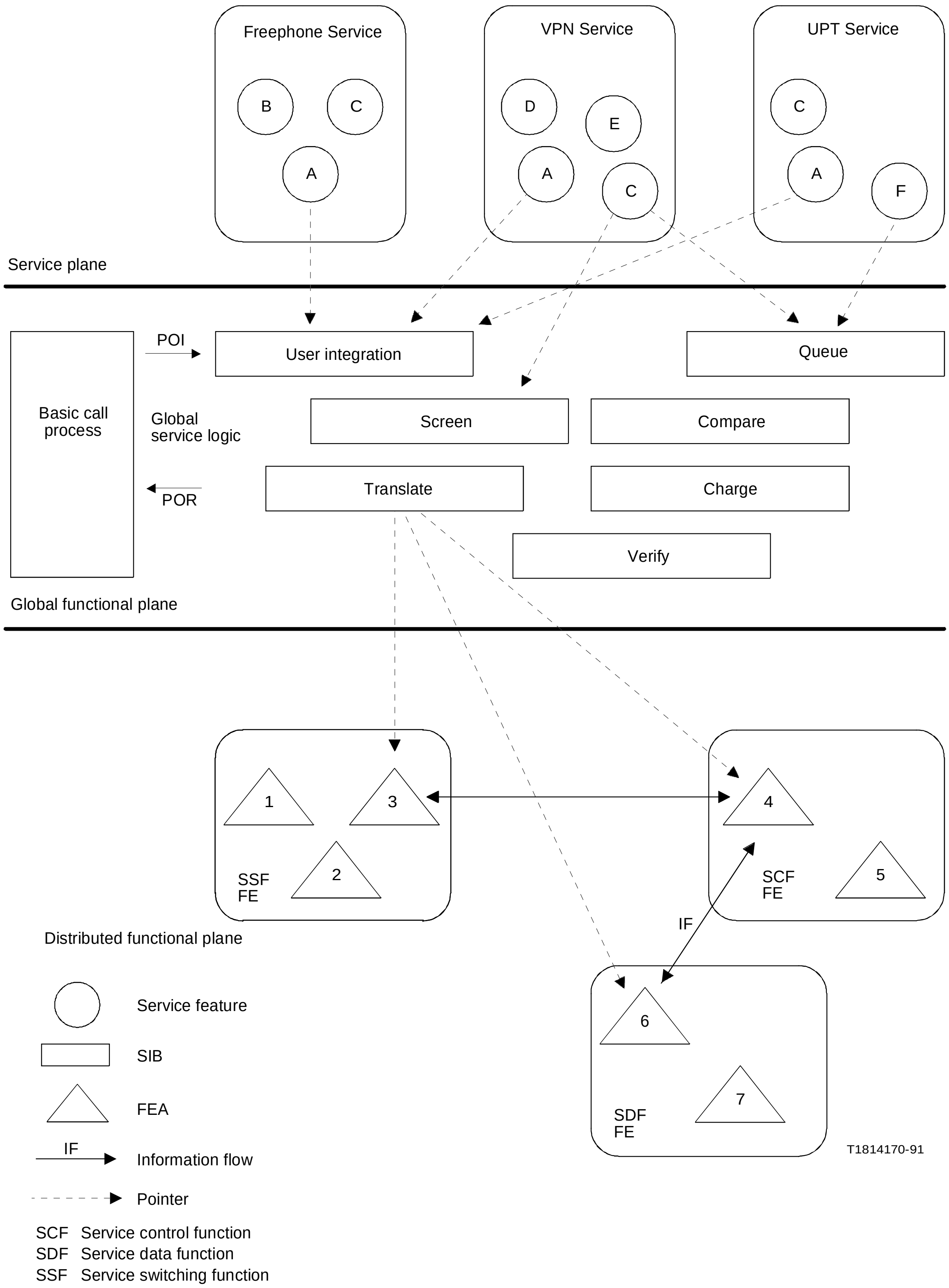


FIGURE 22

Service decomposition

4 Intelligent Network (IN) long-term architecture framework

4.1 *Introduction*

The IN long-term architecture framework is the structure whereby there is integration of technologies developed in other standards activities (i.e. ODP, DAF, TMN, etc.) into the intelligent network architecture. The framework will provide an open architecture that is achieved through the integration of computing information and telephony technologies.

This framework will structure target architecture around the concepts of providing a bridge to existing architecture by utilizing enabling technology, but not being restricted by today’s technology.

The architecture will be enhanced by evolving service requirements and emerging technologies. The emerging technologies should include broadband capabilities, distributed processing (e.g. distributed data base), Open Systems Interconnection, object-oriented modelling, information technology, cooperative processing, distribution control, management of services and network, verification/validation and artificial intelligence.

4.2 *Intelligent Network Conceptual Model*

It is intended that the IN conceptual model remains consistent throughout the evolution of the IN architecture. Evolutionary changes based on experience, service requirements or technologies should be accommodated, as necessary, within this constraint.

4.3 *Architecture structure*

4.3.1 *Logical architecture*

The IN long-term architecture will facilitate the development of intelligent networks. These networks will have the attributes of: integrated services; integrated/shareable control; programmability; adaptability facilitated by modularized hardware and software; interoperability of networks and systems; an OSI-aligned protocol architecture for all interfaces that facilitate communication between entities. The networks consist of interconnected nodes. Each node consists of functional groupings including: service function; interconnect function; and communication function. Service function contains service logic, which includes the capability to translate customer requests into network actions necessary to execute those requests, functional components and service control programmes. The interconnect function establishes and maintains the connections, while providing the network and network node interfaces. The communication function facilitates communications among network nodes/units, among networks, and between the network and the customer. By definition, a node may provide the entire set of functions or any subset.

4.3.2 *Physical architecture*

Elements of the physical architecture are nodes and interfaces. With the expected unification of all protocols and the alignment of the resulting protocol with the OSI reference model, all the interfaces should have the same protocol architecture. That includes the network-to-network interface. However, this does not mean all the interfaces would be identical. It means that one would have to implement, only, those protocol messages necessary to cover the functionality at that interface. The syntax and the semantics of those messages, though, would be the same. The OSI reference model is expected to evolve to accommodate the specific needs of telecommunications.

4.3.3 *Open distributed processing view*

The open distributed processing view of the IN long-term architecture introduces the concepts of application portability and an open system. Modelling an intelligent network or its node as an open distributed information system can be used to facilitate application portability and open communication. Application/information processing can be defined as a service. Interactions between the service and the computing system platform, on which the service runs, is required.

From a computational viewpoint, the service and the computing platform can be represented as aggregated collections of objects. An arbitrary boundary between the service and the computing platform is used to differentiate objects in the service from those provided by the network provider in the platform (see Figure 23). An object should be considered as a set of software/hardware components. The computing platform must be powerful and flexible enough to support the expected service needs. The platform's capabilities will be determined by service needs and technological opportunities.

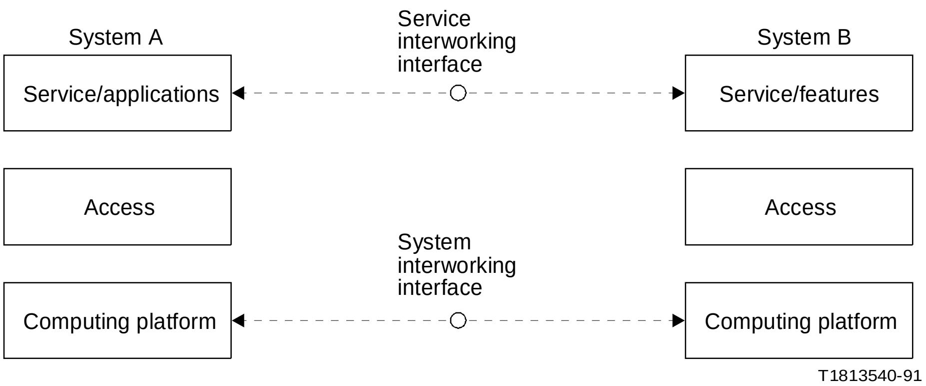


FIGURE 23

Open distributed processing view

Portability of distributed applications guarantees that objects executed by a system have the potential to migrate to another system without disruption of the service provided, or the services used. Migration includes both the static case of reconfiguration and the reloading of a system and dynamic reconfiguration.

Open systems allow a broad range of activities to be accomplished, such as: interprocess communication; data representation; data storage; processing and resource management. Cooperation between open systems is accomplished through two abstract interfaces: service interworking interface and system interconnection interface.

Additional open distributed processing viewpoints are described in the ODP/DAF standards, each representing a different set of abstractions of the distributed system. Viewpoints are pragmatic tools, each leading to a representation of the system with emphasis on a specific concern. Within IN the notion of viewpoints has been adopted in the concepts of planes in the IN Conceptual Model. Multiple viewpoints must be considered to ensure portability.

Distribution transparency is a DAF mechanism that achieves evolvability and compatibility with existing technologies. Distribution transparency refers to the hiding or isolation of the effects of distribution. Transparency may be supported by providing sets of transparent objects within the infrastructure. For example, location transparency hides the location of an object from the other objects that interact with it. Within IN, location transparency can be used to support subscriber portability which enables a subscriber to access a service in the same manner independent of the location from which the network is accessed. Additional transparencies relevant to IN include concurrency transparency, which hides the existence of concurrent users of an object, and replication transparency, which hides the effects of having multiple copies of objects. The integration of transparency into the design of interfaces in the IN architecture enables greater freedom and independence in the design of applications and services. Services may be designed independent of where, within the system, they are implemented and independent of the physical or logical configurations of the system.

4.4 *Service considerations*

It is intended that the long-term architecture support video, image, audio, text and data services, etc.

4.4.1 *Service/service feature interaction*

As part of the service definition, interaction between independently defined services will have to be taken into account. The handling of these interactions will utilize a mixture of technologies described in § 4.5 and new methods to be devised within the long-term architecture framework studies.

Two types of interactions will have to be taken into account:

– interactions resulting from combined usage of service definitions;

– interactions resulting from the interactions of components with which the services are built.

(This will recursively apply for the service components.)

To handle interaction situations an identification of these situations will have to be made. Amongst others, two methods are foreseen:

– static interaction: identification resulting from common usage of resources (data);

– dynamic interaction: identification resulting from sequencing services/components.

Thus, several phases can be foreseen at which interactions should be handled:

– at service definition (creation) level;

– at service deployment level;

– at service activation level;

– at service execution level.

(This can be recursively applied to components building the service.)

A mixture of methods and tools in all phases will be needed to handle the interaction problem.

Service creation, service deployment, service activation and service execution levels will require further study.

4.5 *Technology basis*

It is intended that the IN long-term architecture take advantage of new/emerging technologies as appropriate. Several of these technologies are described in the subsequent §§ 4.5.1 through 4.5.10.

Intelligence resides in all physical and logical elements of the IN network. This distributed intelligence gives the network the flexibility to respond to new requirements in an efficient and effective manner.

4.5.1 *Broadband capabilities*

Broadband transport, traditionally, has concentrated on service control and bearer control functionalities. The support of new capabilities in switching and transport provided by emerging new techniques (e.g. new transfer modes, different communication configurations and the separation between call and connection control) will lead to additional requirements on the IN long-term architecture.

4.5.2 *Distributed processing*

Distributed processing is the mechanism for maintaining a multi-functional environment, that is composed of a variety of applications, protocols and platforms. Several important issues include scalability, portability and performance components.

Scalability and portability are the means to interface with various platforms, independent of size or complexity. In the area of network services, it allows the construction of remote and local operations; utilization of information system resources via applications, where the applications are unaware of whether the interface platform is local or distributed, and that the lack of awareness should not affect their proper operation.

As networking interfaces improve, it will be possible to treat groups of applications as a single image of an application platform, known as distributed systems. A distributed system may have common objects, authentication information and software/service interaction. An important area is data communication services, which function as data transport services.

The concepts of data base management also need to be applied to the network environment. The conventional view of data base management has been applied to data manipulation capability and hardware and software retrieval methods. Data base services are the specialized data services required to access and manage structured elements through the management of information processing and the data base infrastructure.

Standards work in distributed processing focuses on the integration between diverse applications and across diverse technologies that include telecommunications, multimedia systems, and information services.

The seminal work in modelling of distributed systems and architecture for distributed processing within the standards arena are being done in CCITT SG VII, Question 19 DAF (Distributed Application Framework). DAF, in collaboration with ISO/IEC JTC1 SC 21 ODP (Open Distributed Processing), is producing a number of Recommendations (to become the X.900-Series of Recommendations) on a framework for standardization of distributed processing. The primary objective of these efforts is the provision of a standard infrastructure which enables distributed transparency, or the hiding of the effects of distribution from the applications. This transparency includes independence from the details of the communications mechanisms to be used.

A complete distributed processing solution addresses issues regarding interoperability among heterogeneous systems, portability of software between heterogeneous systems, integration of heterogeneous software and technologies to provide consistent solutions to future information needs, scalability of these solutions, and improved performance.

4.5.3 *Open Systems Interconnection (OSI)*

Communication systems which employ the standardized communication procedures and methods derived from the OSI Reference Model and interconnection are referred to as “Open Systems” and “Open Systems Interconnection” respectively. Defined procedures enable the interconnection and the subsequent effective exchange of information between users. The OSI Reference Model, in particular, will permit interworking between different networks, of the same or different types, to be defined such that communication may be achieved as easily over a combination of networks as over a single network.

OSI is concerned not only with the transfer of information between systems, (i.e. transmission), but also with their capability to interwork to achieve a common (distributed) task. In other words, OSI is concerned with the interconnection aspects of cooperation between systems, which is implied by the expression “systems interconnection”.

An integral part of OSI is the concept of a layered architecture, with defined layers, entities, service-access-points, protocols and connections. Elements of layer operation include connections, transmission of data, error functions, routing aspects and management aspects. The management aspects of the OSI architecture are described in § 4.5.8 (Management of services and networks).

The resulting OSI architecture is composed of seven layers: application layer; presentation layer; session layer; transport layer; network layer; data link layer; and physical layer. The functions these layers provide include: communication between open systems; syntax data representation; management and synchronization of dialogue, data exchange interactions and network connections; optimized usage of available network services. The network connections range from point-to-point configurations to complex combinations of sub-networks with different characteristics. The data link connections provide the data circuit interconnection (i.e. communication paths in the physical media).

4.5.4 *Object-oriented modelling*

The use of object modelling could satisfy the modelling needs of IN long-term architecture by the use of abstract object modelling concepts. These concepts are utilized within both the TMN (Telecommunications Management Network) work and the DAF (Distribution Applications Framework) work.

Generic network modelling for TMN is a common set of principles that utilize generally accepted modelling guidelines which facilitate the process of defining and selecting interface protocols. DAF has defined an abstract object model which provides a formal framework.

The object-oriented methodology allows strict modular system specifications. The strong encapsulation supported by object-oriented design is a prerequisite for openness to evolutionary changes within a specification. The realization of the potential benefits that can be achieved by the user of object-oriented analysis and design has resulted in the adoption of this approach in various domestic and international standard arenas.

To ensure the tractability of large distributed systems, they must be designed in a manner that minimizes the interdependence of components of the system. Object-oriented modelling techniques are used because they provide the benefits of abstraction, encapsulation and modularity. Abstraction is a tool for simplifying system descriptions by describing characteristics that are meaningful to users while suppressing the irrelevant characteristics. Encapsulation is a technique for only exposing the observable behaviour of an object's services and providing clients with information about how to invoke these services while hiding the details of the object's implementation. Modularity is achieved because object models permit the specification of a system as an aggregation of collections of objects. Thus, system descriptions are simplified by allowing sub-systems to be treated as independent objects. These tools help reduce the complexity of a system when looking at it from a particular point of view.

The ability to reuse specifications, a virtue in a non-distributed environment, becomes a necessity in a highly distributed one. Specifications must therefore be highly modular and knowledge of components must be restricted as much as possible. Object modelling promotes modularity by enabling the structuring of specifications into smaller parts. By constraining all object interactions to take place at well defined interfaces, the interdependence of objects is minimized. This provides a basis for reusability and adaptability. By isolating and explicitly describing all object interactions, the dynamic and evolutionary nature of distributed systems can be more easily modelled.

The object-oriented design philosophy enforces a discipline and precision of specification that is required for the description of complex, distributed systems. For the IN long-term architecture, there is the need to outline an abstract object model and to begin using this paradigm for the description of objects and interfaces required in the IN long-term architecture.

4.5.5 *Information technology*

Information technology provides the means for an enterprise to integrate, share, process and distribute large quantities of information in order to meet the needs of a wide variety of internal and external customers. System design, software development, and operation procedures are important aspects of this technology. In the context of IN, this technology would initiate transport services; operation, administration, maintenance and provisioning (OAM&P) services; and customer data interactions.

4.5.5.1 *Software/programming assistance*

Software tools, techniques, languages and procedures should play a major role in areas such as service interaction. Independent components of the service interaction may be maintained or initiated by a specific software tool/procedure or a set/series of tools/procedures.

Computer aided software engineering (CASE) technologies enable the management of the entire software development process, from requirements to design, coding, testing, and maintenance. Use of CASE technology is expected to reduce the incremental cost of new application development and to enable new applications to be developed faster.

CASE tools are used to make products that are stored in a CASE data base repository, which provides support for the storage, retrieval, version management, and configuration management of these products and the relationships among them. CASE integrates these various products to ensure consistency among them and to permit reuse at each stage. The repository supports distributed access to these software products and heterogeneous and distributed storage. CASE technology enables the speedy development of new applications by encouraging the practice of reuse. Use of CASE is aimed at achieving an improvement in quality and improving the management and productivity of the process. Existing and new CASE technologies and techniques can be applied within IN to address the dynamic aspect of service creation. The use of CASE technologies within IN will enable the rapid creation and activation of new application components.

4.5.6 *Cooperative processing*

Cooperative processing is a specialization of distributed processing across two systems. Cooperative processing is the means by which both data and network functionality can process cohesively within the processing structure. This is accomplished by the introduction of an operational control mechanism. The operational control components include node connection control, global connection control and service control. Node connection is concerned with the manner in which a connection is established through an individual network node to a specific link which leads to another node (i.e. circuit-based or packet-based connection). Global connection is concerned with the manner in which one or more end-to-end connections through multiple nodes is established and monitored in the network. Service control provides the ability to effectively define and customize a service.

The control components of IN consists of those functions and activities through which distributed network intelligence is coordinated, administered, and utilized. The control intelligence is itself distributed throughout the network to allow for the necessary scalability and performance. New services which utilize the intelligence throughout the network will require a more robust control model.

The availability of remote data base(s) will allow the introduction of new services, which will need access to large amounts of volatile data. Therefore, the types of services that can be defined and employed in the network could be easily expanded. Remote data base(s) may provide "intelligence", in the sense that the specific data returned as a result of a retrieval request may depend on logic embedded in the data. This is not "traditional" network control functionality, in the sense that something other than the requested information may be obtained. It may, however, be considered to be a part of service control. If the data base logic is used for only one service then this control can be considered to be explicit. If it is used for multiple services, then the control will become implicit. In either case, most of the control will still be embedded in, and integrated with, the node processing function.

Another example of cooperative processing is the utilization of "intelligent terminals", e.g. a workstation consisting of a "smart card reader", which performs some of the network tasks (i.e. authentication and authorization).

4.5.7 *Distributed control*

A major part of distributed control is the allocation, control and management of resources. Certain types of resources can be centralized and other types can be distributed. Distributed control makes the availability of resources more effective. Connection to alternative pools of resources would also be possible. The function of reserving resources is also important. There could be a need of a specific resource type for certain services to be reserved before establishing a connection or replying to an application request.

Distributed control also provides the framework for representing and handling service/feature interactions.

High-level networking services provide an application with a very high-level interface to networking capabilities. These services do not require the application to be aware of any of the low-level network details.

Simple networking services are designed to be used to provide applications with an interface to simple application-to-application communication over a network. This interface will be used in applications that perform explicit connection-oriented or connectionless networking activity, but do not want to control the low-level details of the networking interface.

4.5.8 *Management of services and networks*

The management aspects of IN can be summarized as the management of service and network generic functions, along with IN specific managed objects. Managed objects are very diverse. They can be divided according to their logical, functional and physical nature. Examples of managed objects include: UPT service, service independent building blocks, trigger table data, network services (i.e. LAN, ISDN), transmission services, network operator role, OSI application process, OSI resources, virtual circuits, connections and calls.

To manage all the different kinds of managed objects, an integration of the IN architecture and TMN concepts should be accomplished. One of the outcomes of this integration will be the identification of IN specific management requirements concerning the deployment, provisioning, control, monitoring and billing/charging of services. These requirements contain amongst others configuration, performance, fault, accounting and security aspects.

The network environment needs to manage physical connections, network protocols and formats, and distributed systems services.

Within the OSI architecture, there is a need to recognize the special problems of initiating, terminating, and monitoring activities and assisting in their harmonious operations, as well as handling abnormal conditions. These have been collectively considered as the management aspects of the OSI architecture. The defined categories of management activities include application management, systems management and layer management.

Because of the vast programmability possibilities of new services, the IN long-term architecture should also support the ability to program (in parallel with the service processing logic) management logic. To accomplish this, management SIBs may be provided.

4.5.9 *Verification/validation*

The function of verification/validation is to insure that features/service interactions perform as specified. This may include testing of network performance to ensure consistency with Quality of Service and network integrity objectives. This can be accomplished by providing complimentary verification/validation standards to qualify functional performance and consistency.

4.5.10 *Artificial intelligence*

Artificial intelligence is the means by which IN will evolve the existing network base with the latest emerging technologies. The concepts utilized may involve non-traditional network elements. This should facilitate a wide variety of services and interfaces (i.e. user-to-user, user-to-network, network-to-network, etc.).

ANNEX A

(to Recommendation I.312/Q.1201

Alphabetical list of abbreviations used in this Recommendation

API Application programming interface

ATM Asynchronous transfer mode

B-ISDN Broadband-ISDN

BCP Basic call process

CASE Computed aided software engineering

CS Capability set

DAF Distributed application framework

DF Distributed functional

DFP Distributed functional plane

DSL Distributed service logic

EF Elementary function

FE Functional entity

FEA	Functional entity action
GF	Global functional
GFP	Global functional plane
GSL	Global service logic
GW	Gateway
IF	Information flow
IN	Intelligent network
IN-SL	IN-service logic
INCM	IN conceptual model
ISDN	Integrated services digital network
LAN	Local area network
LTCS	Long-term capability set
N-ISDN	Narrow-band-ISDN
OAM&P	Operation, administration, maintenance and provisioning
ODP	Open distributed processing
OSI	Open Systems Interconnection
P	Protocol
PE	Physical entity
PLMN	Public land mobile network
POI	Point of initiation
POR	Point of return
POTS	Plain old telephone service
PSPDN	Packet switched public data network
PSTN	Public switched telephone network
SCF	Service control function
SDF	Service data function
SF	Service feature
SIB	Service-independent building block
SMF	Service management function
SMP	Service management process
SSF	Service switching function
STM	Synchronous transfer mode
TMN	Telecommunications management network
UPT	Universal personal telecommunication
VPN	Virtual private network