ETSI TECHNICAL REPORT

ETR 061

February 1993

Source: ETSI TC-NA

Reference: DTR/NA-041102

ICS: 33.080

Key words: ISDN

Integrated Services Digital Network (ISDN); Rules for the construction of reference configurations

ETSI

European Telecommunications Standards Institute

ETSI Secretariat

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

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

Contents

Foreword

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

This ETR has been produced by the Network Aspects (NA) Technical Committee of the European Telecommunications Standards Institute (ETSI) and provides the values for the construction of Reference Configurations (RCs).

This ETR describes the RC for Integrated Services Digital Network (ISDN) telecommunications systems. This ETR describes the objectives of RCs, details the methods used for RCs and outlines the application of rules and principles for the generic RC and an example of a specific RC.

Introduction

The RC is now generally accepted as a useful and important tool for providing a referential description of ISDN telecommunications systems, both generic and specific, and in particular for B-ISDN. This is achieved first, through the use of widely accepted standard Functional Groups (FGs) and Reference Points (RPs), which retain their meaning within more detailed system development, and second, to the generic nature of the RC which can be used as the source of many specific realisations.

There is still, however, the need to precisely define a RC, and its properties. This is partly because there is no independent definition of FGs and RPs (at present a circular definition is used), and partly because after its referential use, it is usually developed to form a functional description of a specific system realisation. Without its generic properties, therefore, it can no longer be called, or used, as a RC, but instead becomes a functional realisation or architecture. RCs can, therefore, sometimes be misused, through confusion with the functional representation of specific system realisations.

A RC depicts FGs and RPs which are necessary to fulfil a specific task. A RP can be defined by the data being transferred between FGs, and a FG can be defined by the mapping of data between RPs. It is not concerned with the location or storage of this data (other than the local data within the FG). RCs are functional models at the conceptual level, concerned with arrangements of the FGs and RPs to anticipate possible physical realisations. When physical realisations are described, these may include FGs, interfaces and RPs. The presence of at least one interface indicates that a physical configuration is being described. Two types of functional mappings from conceptual to physical arrangements are envisaged. The first is where the FGs are physically separated, and will require an external interface between them for communication. The second is the mapping of functions into the internal parts of physical entities, when internal interfaces of software or hardware will separate them. At the design stage, it may not be clear which of the two types of mapping should be applied to each function and RP.

Because the RC is such a widely used and useful tool, a number of issues must be considered before the most suitable description using them can be obtained. These are described in Annex A, and include meaning of different levels of abstraction, representation syntax, and communication between levels. It may not be possible, nor desirable, to standardise all of these issues; to do so may require so much formal description as to make the resultant descriptive language difficult to use. A clear statement to accompany a RC will help the user obtain a clear understanding of what is intended. It should still be possible to have a user-friendly, useful and precise-enough description to support RCs, and the following sections attempt to provide such a basis.

Since the RC can be used at an important early design stage, it is necessary for it to be able to be flexible, and to describe large systems, in an understandable way, chosen by the designer. Four important concepts, therefore, are used:

1) system partitioning

This means that the total system can be represented by a number of related RCs; an example would be the representation of the B-ISDN by three planes (user, control and Network Management (NM) [1]);

2) a family of RCs

Existing at different levels of abstraction. Lower levels give increased detail about the level above, with each complete level having the same overall properties as the level above;

3) a family of RCs

Existing at different layers which take into account the Open Systems Interconnection (OSI) layering of functions. All layers are needed to describe the overall behaviour in this case;

4) a family of RCs

Existing at different strata (i.e. taking into account the stratification concept).

All of the above concepts allow the original system to be simplified by considering only part of the overall system at any one time. However, the rejoining together of the separated parts should be made easy to achieve, and this requires care in defining the partition boundaries.

An example of the use of the principles in this ETR to depict a RC for the B-ISDN showing all the planes is given in figure 1.

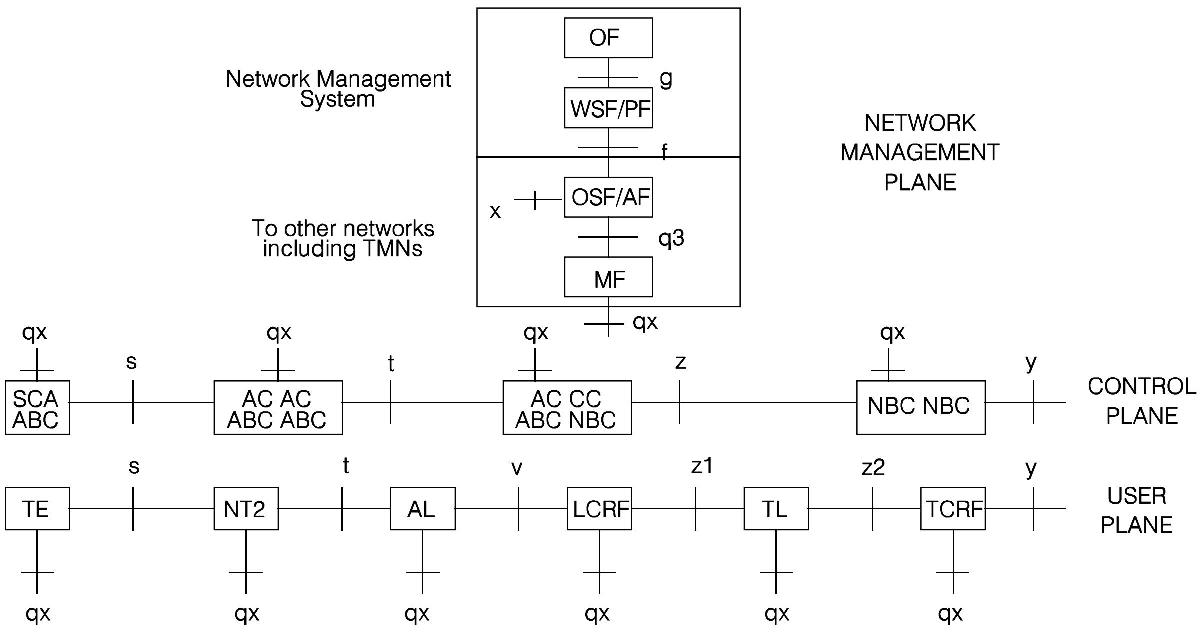


Figure 1: A generic RC for the B-ISDN showing the user control and NM planes

1) The rules described in this ETR apply mainly to the user and control planes. Possible modifications and additions for the NM plane, especially regarding information modelling, is for further study, see subclause 5.6.

1 Scope

This ETSI Technical Report (ETR) describes the Reference Configuration (RC) for Integrated Services Digital Network (ISDN) telecommunications systems.

This ETR describes the objectives of RCs, details the methods used for RCs and outlines the application of rules and principles for the generic RC and an example of a specific RC.

Annex A gives information about the concepts used in this ETR and a specification of functions and messages is given in Annex B.

2 References

For the purposes of this ETR, the following references apply.

[1] CCITT Recommendation I.324: "ISDN network architecture".

[2] CCITT Recommendation I.325: "Reference configurations for ISDN connection types".

[3] CCITT Recommendation I.340: "ISDN connection types".

[4] CCITT Recommendation I.327: "B-ISDN functional architecture".

[5] CCITT Recommendation I.130: "Method for the characterization of telecommunication services supported by an ISDN and network capabilities of an ISDN".

[6] CCITT Recommendation I.310: "ISDN - Network functional principles".

[7] CCITT Recommendation M.3010: "Principles for a telecommunications management network".

[8] CCITT Recommendation I.320: "ISDN protocol reference model".

[9] CCITT Recommendation I.321: "B-ISDN protocol reference model and its application".

[10] CCITT Recommendation Q.65: "Stage 2 of the method for the characterization of services supported by an ISDN".

[11] CCITT Recommendation X.210: "Open systems interconnection layer service definition conventions".

3 Definitions and abbreviations

3.1 Definitions

From CCITT Recommendation I.324 [1]: "RCs are conceptual configurations which are useful in identifying various possible arrangements in an ISDN. The RCs are based on association rules of functional groupings and RPs".

Clearly, the only constituents of RCs are FGs and RPs. Association rules between these are given in Clause 5. The arrangements of the FGs and RCs can anticipate possible physical realisations. The presence of at least one interface indicates a real physical configuration, and not a conceptual configuration.

The RC depicts FGs and the flow of data between them. It is not concerned with the location or storage of this data (other than the local data within the FG).

From the foregoing discussion, the following updated definition for B-ISDN is proposed:

RCs are conceptual configurations which are useful for identifying possible arrangements in a B-ISDN (Integrated Broadband Communications (IBC) network). The RCs are based on association rules of functional groupings and RPs.

The properties assumed from the above is that RCs can be composed of well known and understood components (possibly drawn from a widely accepted standard list), and that RCs can be highly generic, i.e. capable of representing many possible arrangements and realisations.

To fully exploit the above definition, the following uses are implied:

1) to identify referential locations for the system (or part system) being developed;

2) to identify possible arrangements (and developments) having the same properties as parts of the (generic) RC;

3) to provide a description of the behaviour at the conceptual level so that the equivalence of the possible arrangements can be determined.

There appear to be two possible problem cases:

1) where a non-conceptual description is intended (of actual hardware and software) but describing parts of the system with FGs and RPs where no interfaces occur or have yet to be selected;

2) where a specific implementation is being described using one or more FGs or RPs not widely accepted or known. These cases should not be described as RCs because they do not have the properties required.

3.2 Abbreviations

For the purposes of this ETR, the following abbreviations apply.

ABC	Access Bearer Control
AC	Access Control
ACC	Access Connection Control
AL	Access Link function
ATM	Asynchronous Transfer Mode
B-ISDN	Broadband ISDN
CC	Call Control
CC	Connection Control
CDRP	Context Dependent Reference Point
CE	Connection Element
CFRP	Context Free Reference Point
CT	Connection Type
FC	Functional Component

FE	Functional Entity
FG	Functional Group
IBC	Integrated Broadband Communications
ISDN	Integrated Services Digital Network
LCRF	Local Call/Connection Related Function
MF	Mediation Function
NBC	Network Bearer Control
NCC	Network Connection Control
NEF	Network Element Function
NM	Network Management
NSF	Network Specialised Function
NT	Network Termination function
NT2	Network Termination 2
O&M	Operations and Maintenance
OF	Operation Function
OSF	Operation System Function
OSI	Open Systems Interconnection
QOS	Quality of Service
RC	Reference Configuration
RP	Reference Point
SCA	Service Control Agent
SDL	Specification Description Language
SFPL	Specialised Function Provision Link
TCRF	Transit Call/Connection Related Function
TE	Terminal Equipment function
TL	Transit (national) Link function
WSF	Work Station Function

4 Objectives of reference configurations

Although RCs have been in use for a considerable time to describe an arrangement of FGs and RPs which is necessary to fulfil a specific task, it is necessary to provide a clear description of the purposes of RCs. The following objectives will, it is hoped, provide this need.

The major objectives of RCs are:

- to provide RPs within the RC which can be used to help define network performance measurements, domains of responsibility of service providers and network operators, as well as physical interfaces between equipments and systems;

- to anticipate possible physical relationships between the realisations of conceptual FGs used to provide services to a user. Both telecommunication and management services are included. As a result, physical realisation options for providing these functions should be more easily determined;

- to provide generic reference models which can be used as the source for many alternative equivalent physical realisations which possess the same overall functional behaviour. Also, such models can be used to show the fundamental similarities between families of apparently different configurations;

- to provide a topological model at different levels of abstraction;

- all levels should have the capability of being implemented, i.e. capable of being transformed into specifications of hardware and software;

- to help discover and specify standardised interfaces in terms of the data and messages passing across them;

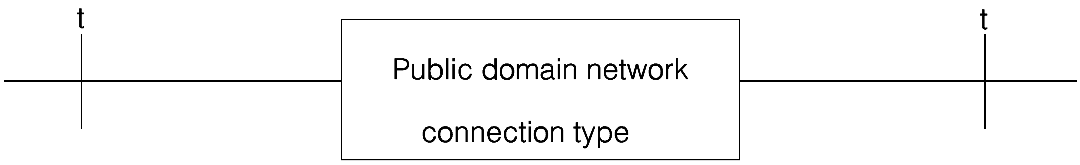
- the most abstract levels should be generic, i.e. largely independent of implementation options (such as technology, economic and regulatory constraints), and be stable enough with time so as to be used as a basis to describe evolutionary changes;

- to provide the link between the conceptual function description and the physical realisation;

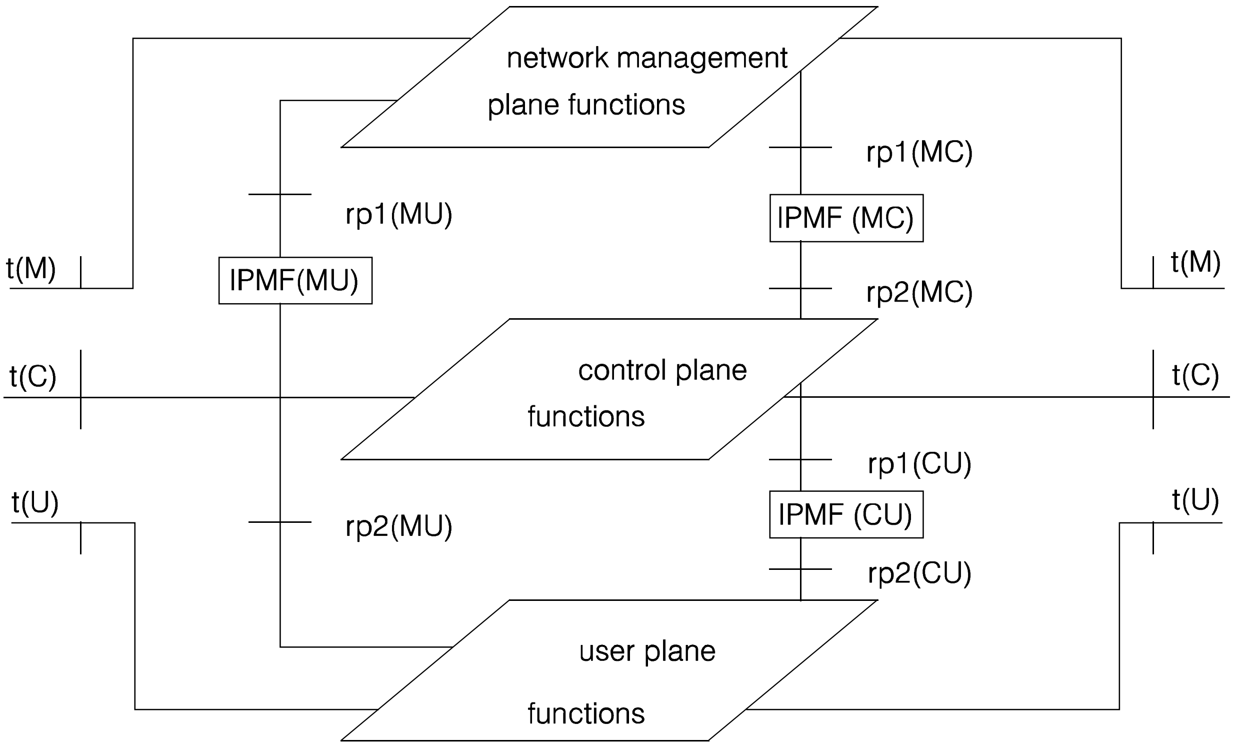
- to identify referential locations for a system (or part system) being developed.

5 Detailed description of the methods

In general, it is necessary to deal with the description of large systems in parts. One method by which the overall system is partitioned is to create planes separated by RPs, see figure 2. It may be convenient for these RPs to characterise a "service" between the constituent parts.



a) First stage of development showing most abstract description in terms of connection types.



b) Next stage of decomposition into three planes with bridging reference points and interplane management functions to support a selected connection type.

Figure 2: Development of RCs for a given connection type

However, it is important to correctly describe what service can be accessed at the point described.

A telecommunications service is characterised using the CCITT Recommendation I.130 [5] 3-stage method as follows:

- Step 1.1: prose service definition and description;
- Step 1.2: static description of the service using attributes;
- Step 1.3: dynamic description of the service using graphic means.

Together, these three steps define the service characteristics as they apply, from the user's point of view, at a given access RP where the service is accessed.

In the general case, the use of steps 1.1 to 1.3 above can be applied at any junction between two functions communicating by messages.

A "service" can be viewed from either the user's or the provider's viewpoint. Particularly when a human user is involved, these two aspects can differ. Network performance parameters describe those parameters of the behaviour of a network which are only visible and quantifiable to the provider of that service. The term service performance can also be used to describe the service offered. Quality of Service (QOS) describes those aspects visible to the user or consumer of the service.

Methods of dividing the overall system into separate "service" regions in this way are described below. RCs which describe the supporting FGs required for different services, available at RPs, can be found within each of the parts. Each RP and RC is assumed to possess attributes which are used to characterise the service supported.

5.1 Partitioning of reference configurations

During the development of a RC for a large system it may be necessary, for convenience or other reasons, to consider only part of the whole RC. An example of this could be the need to consider the user, control, and NM activities separately as planes. To maintain completeness of the whole description, it is then necessary to introduce bridging RPs to describe the data flow between the parts and plane management functions to translate between the planes. This is illustrated in figure 1, in which the public domain network is partitioned into three planes, user, control and NM. The relationship between these planes and those used in CCITT Recommendations I.320 [8] and I.321 [9] will need to be resolved.

For the simplest type of partition, the three planes use the same descriptive framework, and single RPs can be used to separate each pair of planes; an example showing the separation of the NM plane is shown in figure 1. However, when it is desirable to use different descriptions to allow different developments to occur in each plane with minimum of impact on the other planes, then it will be necessary to introduce a function which performs the separation of the planes. This function is referred to as a plane manager, as used in CCITT Recommendation I.320 [8], although inter-plane manager is a name which is more illustrative of the function required. One inter-plane manager can be introduced between each pair of planes. In this case RPs rp1(UC), rp2(UC), rp1(CM), rp2(CM), rp1(MU), and rp2(MU) are introduced to describe data flows between the partitioned planes as shown in figure 2.

Further subdivision of each of the parts may cause the bridging RPs to become distributed (as in figure 1). Clearly, partitioning will only simplify the overall description when the parts are nearly or completely disjoint.

5.2 Levels of a RC

The RC may be decomposed into levels which give increased detail of the composition of the levels above. The properties of such a model are given below:

a) each level of the RC should be self contained, i.e. it should not be necessary to refer to lower levels to obtain an understanding at a given level;

b) both FGs and associated data may be decomposed into more detailed descriptions, with both functions and data retaining matching relationships to the levels above;

c) usually, each level of decomposition gives increased detail about the level above, with each level having the same overall properties as the levels above. The generic property at lower levels, however, will diminish due to exercise of options.

Unfortunately, no generally agreed definitions of levels exist at present. For example, levels can be implementation, technology or country independent; they could represent all or only part of the level above. Furthermore, at lower, (more detailed) levels, the description becomes a specific realisation with possibly little generic content. In this case, the use of the term RC to describe this system is not appropriate. More specific descriptions, such as physical realisation or architecture, should be used. The exact level of detail at which this transition occurs may not be easy to identify, and indeed, may change with time.

Hierarchic levels may exist within each of the three partitioned RCs (user, control and NM planes).

5.3 Layers of a reference configuration

In a layered RC, it is usual to represent on a lower layer only the services provided to the layer above. In this case, therefore, the overall properties of the system require a consideration of all layers. Communication occurs between peer layers of separated layered groups; it also occurs between adjacent layers within a group. The ISO seven-layer system would be an example of such a configuration.

5.4 Stratified reference configurations

A stratified RC can be used to describe the stratified reference model. Details are for further study.

5.5 The role of reference configurations in an overall system specification method

A general method of development for characterising complete telecommunications systems in general in which RC play their part particularly in stage 2, is given in the CCITT Recommendation I.130 [5]. This is a three stage method in which:

a) for a given plane, the service to be supported by the RC is characterised from the user's viewpoint (stage 1);

b) a functional model of the network which supports the characterised service is generated, comprising a set of Functional Entities (FEs) independent of physical location (or hardware or software), with communication relationships between them in terms of data or message flow. CCITT Recommendation Q.65 [10] gives more detail of this step. Stage 2 gives details of steps 2.1 to 2.4;

c) a set of conceptual FGs which are anticipated to be separated physically are produced, normally these arise from the user plane. With RPs these form the RC. This is not explicitly stated in CCITT Recommendation I.130 [5] description;

d) the FEs within b) are mapped onto the conceptual FGs of the RC in c). Some FEs will be spread across more than one FG. Some FEs will exist within an FG with internal interfaces. Full details of message communication between the FGs can now be finalised, thus defining the RPs. Stage 2, step 2.5 at the conceptual level only;

e) at this point, the RC production stage has now been completed, but the 3-stage method proceeds to define interfaces.

The development of RCs should take account of this approach, however, in the case of NM the services provided and the structures dealt with may be different from those described for the user plane. Thus, the methodology is likely to differ, especially with respect to information modelling.

5.6 Extensions to the reference configuration to represent modelling with objects

The NM plane differs from the user plane in the type of activity undertaken. Instead of providing user-to-user transport services, the NM plane needs to monitor, decide appropriate action, and implement control of the user and control planes, as well as offer NM facilities to users as services. To reflect this different activity, it is necessary to extend the functional RC model to include more details of the data and entities which are being operated upon by the FGs within the model. This is because the location of management information (to be realised as a common database of some type), as well as the location of the FGs, could be important in deciding the architecture. A similar requirement exists for Intelligent Networks (IN) which are closely related to control plane activities. It should be mentioned that the use of these enhanced techniques in one plane can co-exist with the more conventional models in the other planes. However, certain partitions, such as external schema, can only be used when there exists a global model at some level which links all planes together.

Object oriented methods have been used successfully for system specification and design, particularly in software systems. At the conceptual level, the objects can represent the managed resource, and also the totality of the destinations of messages from the application functions. This aggregation is termed an external management schema. Another use of the object is to represent an entity which is largely "self contained", and is very suitable for software implementation and re-use. It is also suitable for representing semi-autonomous units for (open) distributed processing. In this respect, the object possesses the characteristics of the FG, namely, anticipated (physical/software) separation with access through a standard interface using a set of messages. Indeed, functions as used in the RC may be regarded as particular types of object.

Although the main use, at present, of objects within the NM area is to present a view of the managed resources and of management information, the use of objects to represent FGs within the NM RC is likely to follow later.

The implementation of a NM system description using objects may use a virtual replica of the managed system within the managing system. This approach fits well with the object modelling concepts in which they are used as a "substitute" for the actual managed resources. The advantages are that the proprietary links between the objects and the managed resources are hidden from the managing system; however, real-time constraints in maintaining a proper updated virtual replica in a realisation could be difficult to meet.

The conventional RC for NM contains only FGs and RPs. The extension to represent objects can be achieved by associating a set of accessible objects with each FG. This combination of a FG and objects is called a block, or a conceptual block where this is not already clear. This model is now used in CCITT Study Group IV.

Although the same class RPs are used, there will be additional information flow on account of communication between the objects.

The exact form of modelling required to represent objects in this way is for further study.

5.7 Reference configuration construction rules

The following rules are meant to be satisfied by RCs which are produced to meet the objectives described within this ETR. Non-compliance should indicate that some aspect of these RCs could be misunderstood:

1) at a given layer or level, there is always one and only one RP between two FGs. As a corollary, it follows that at a given layer or level, there is always one and only one FG between two RPs;

2) a RP can be defined by the FGs which are separated by the RP, and by the messages passing between them (stage 2). The messages and their descriptive attributes may be grouped, for example, into levels or planes, so as to match the structure of the FGs;

3) a FG is defined in a similar way to a function, namely, as a state dependent mapping between the domain (ingoing) and range (outgoing) messages, see Annex B. If there is no change in value between at least one of the input and output message attributes, or the input and output states of the function, then the FG is null. For convenience, consideration can be limited only to those attributes undergoing change. A FG, by implication, is composed of two or more functions;

4) physical interfaces which do not correspond to a RP will not be the subject of standardisation;

5) when a FG is defined by only two RPs then these RPs must be different, i.e. comprise different attribute values;

6) in a closed system, every FG must have at least one RP attached to it;

7) the minimum number of FGs and RPs are used in a generic RC to provide a description at each level. Multiple similar parts of a RC are usually represented by one typical member;

8) no particular physical implementation is implied by RCs; spatial relationships, for example, are depicted within physical configurations.

5.8 Approach in CCITT

The CCITT work on RC modelling is described in CCITT Recommendations I.324 [1], on ISDN network architecture, I.325 [2], on RCs for ISDN connection types, and I.327 [4], on B-ISDN functional architecture. Also of importance is CCITT Recommendation I.340 [3], on ISDN connection types, in which attributes relating to connection elements and connection types are defined.

It is intended that the RCs produced as a result of the methods described in this ETR will be compatible with those produced using concepts described within CCITT Recommendations.

As has already been described, the generic nature of the Connection Element (CE) has meant that the basic descriptive material prepared for the ISDN case is directly applicable to the integrated broadband communications network in the user plane.

Extensions are required within the Call Control (CC) plane for the services up to level 7 in the B-ISDN.

In the RC within the NM plane, the basic structure of the CCITT M.30 series of Recommendations has been used, but there is likely to be considerable extension of the operations system functions embracing activities currently performed by the operator, so that a suitable management system will be available for operation within a multi-service, multi-operator environment. In addition, there may be modifications and additions for the NM plane, especially regarding information modelling.

6 Applications of rules and principles

6.1 A generic reference configuration

Figure 1 shows a generic RC of the B-ISDN with all three planes. It has the following generic properties:

- represents all known topologies for the link networks in the user plane;
- represents all conditions in which call related functions are grouped together, and possibly cases where these are distributed;
- a general representation for the control plane;
- a general representation for the NM plane.

At the present time, a number of the constituent FGs are identified by name only, and require the next stage of a description in terms of the mapping of the input to output data, as described in Annex B.

6.2 A reference configuration for a given connection type

A general procedure for producing a RC for a given Connection Type (CT) can be described using the public domain network as an example.

As a starting point, a given CT is chosen between RPs ("t" RPs in this example), see figure 3 [2]. CTs exist at present for the ISDN, and are used to describe the capability of a connection established between two RPs. At present, there exist at least thirteen attributes used to describe a CT. These include: information transfer mode and rate, supported payload types, connection establishment, symmetry, connection configuration, structure, channel rate, connection control procedure, information transfer coding protocol, network performance and interworking, and operation and management. Although not all of these are technology and implementation independent, it is possible to proceed initially with a generic top-down procedure by next partitioning the system into three planes with interconnected RCs as shown in figure 3, and which has already been shown in figure 2. This decomposition is chosen because these planes are largely disjoint, and the description of the overall system is simplified by considering each separately, although bridging RPs and possibly inter-plane management functions may be inserted as shown in figure 1 b). In this case the complete description will then comprise the three RCs, one for each plane, with bridging RPs, and possibly inter-plane management functions.

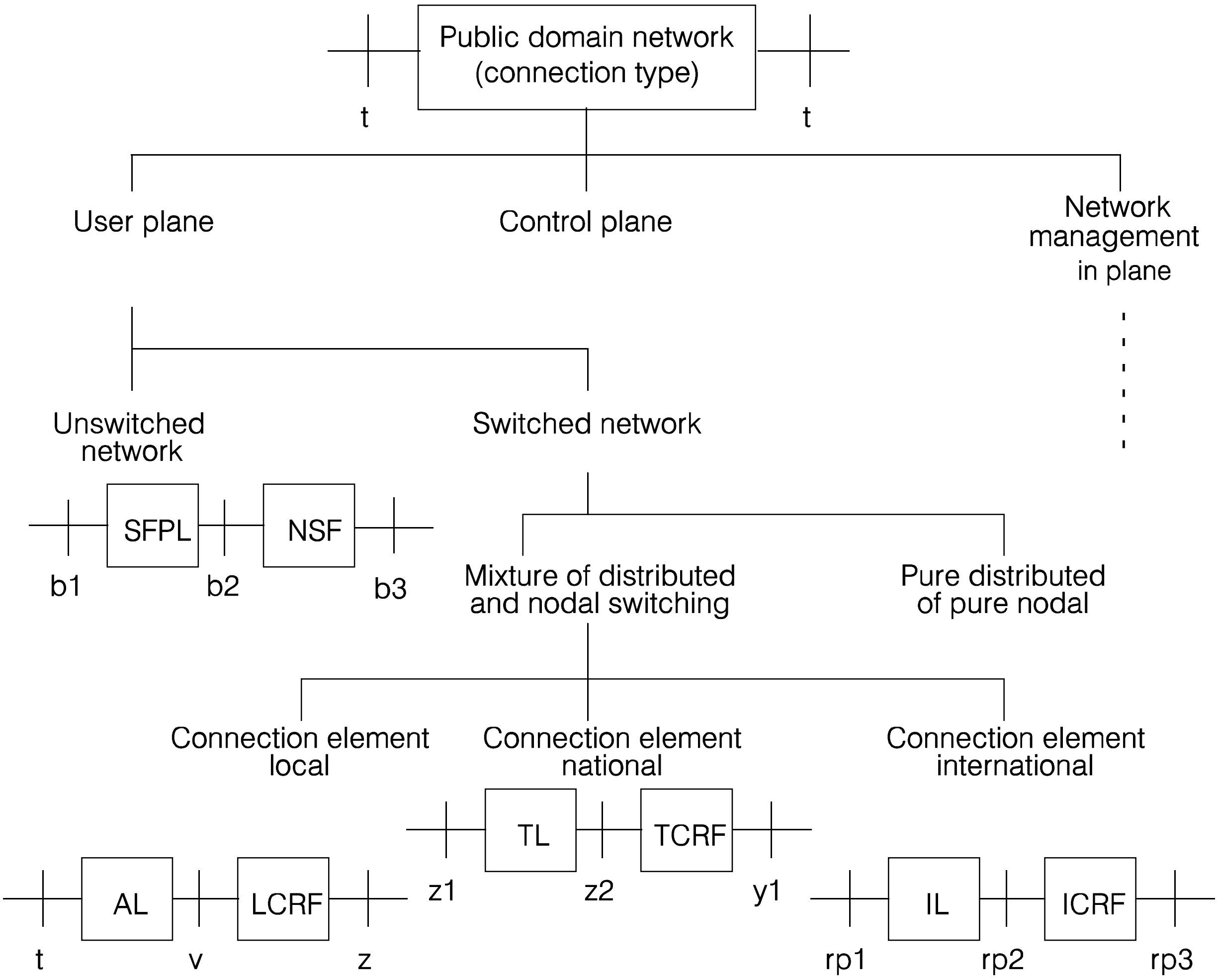


Figure 3: Development of a RC in the user plane, leading to a range of connection elements which will make up a given CT, and showing some of the options available

2) The exact level at which the splitting into planes takes place is for further study since it is not clear at present where this should take place, or if the level will be the same in all cases. The structure shown in figure 3 is there for illustration of the principle only.

Subsequent decompositions can describe the different levels or layers in more detail, each described in as generic a form as possible (i.e. parents of as many as possible subsequent options).

In this example, the next choice for the user plane, shown in figure 3, is whether unswitched or switched networks are required, and then whether a mixture of distributed and nodal, or pure distributed, or pure nodal, switching is required. For the public network, a mixture will be required.

The next development involves the use of CEs in the user plane, as described in CCITT Recommendation I.340 [3]; these provide a very generic support for all CTs, and all known network topologies, including mesh, star and ring. At the highest generic level each CE comprises a link and associated communication related function. Based on signalling and associated address domains, these can be local, transit or international. These are expected to support every CT, so the RC this far is still highly generic. For unswitched networks, no similar signalling separation criterion exists, but (by analogy with the access, transit, etc.) a generic CE exists containing a Specialised Function Provision Link (SFPL) and Network Specialised Functions (NSF). This element can be used, for example, as an unswitched CE in a distribution network.

Further development of each part of the CE then continues: the access link may have shared use of access communication facilities, giving rise to multiplexing and branching functions. The Local Call/Connection Related Functions (LCRF) switching FG may be:

a) non-distributed;

b) partially distributed, resulting in LCRF2 and LCRF1 functions; or

c) fully distributed, involving ring topologies.

In turn, the non-distributed LCRF FG may involve switching, cross-connect, or switching and cross-connect, etc.

The decomposition can continue in this way using established FGs. This procedure essentially involves two activities:

1) selecting options from the general descriptions contained in the FGs in the level above, as already described;

2) choosing particular instances (i.e. technology specific examples) such as an Asynchronous Transfer Mode (ATM) dedicated multiplexor. Decomposition can continue until atomic (i.e. non-divisible) functions are encountered.

Application of the techniques to virtual path and virtual channel connections are for further study.

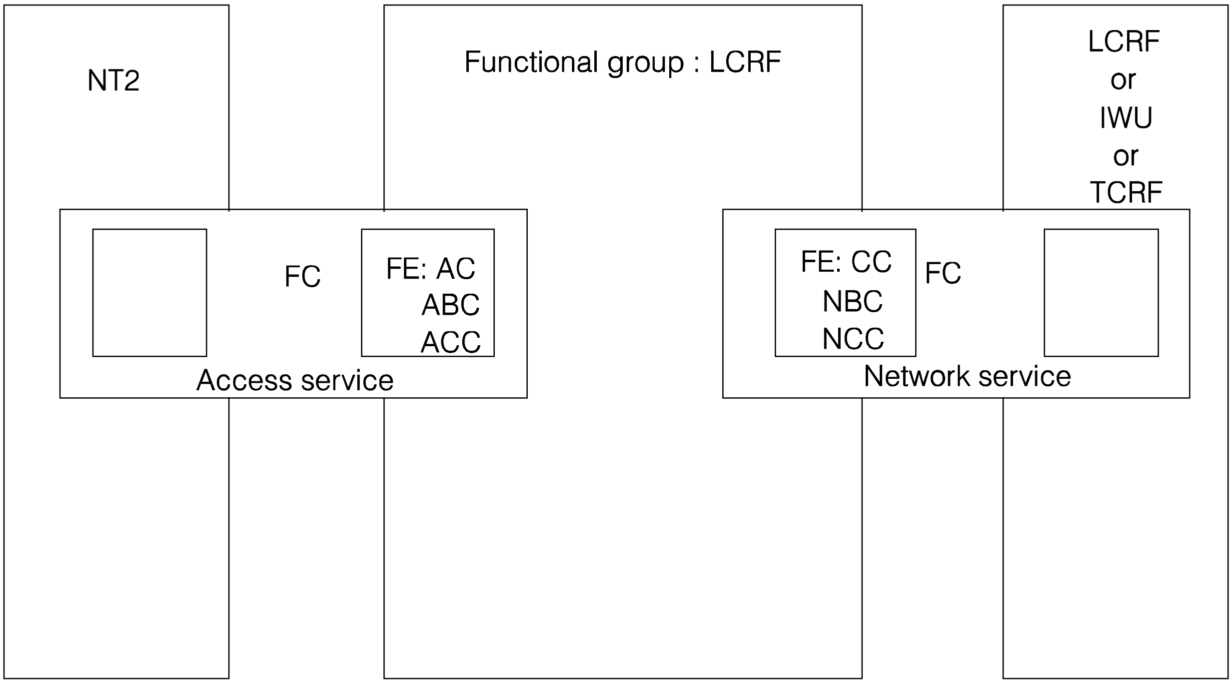
In this example, the level shown in figure 3 can probably be developed a further stage without describing a specific realisation. Subsequent development after this would be a specific realisation.

At each stage, decomposition of the RC should always seek to be as generic as possible (i.e. to represent as many lower-level implementation options as possible). Of course, the number of options still represented by a "generic" RC become fewer as the level of decomposition increases.

The procedure described has shown how the arrangements developed are related to, or derived from, the referential FGs and RPs.

6.2.1 Call control plane

The stage 1 service characterisation is more in line with the Functional Component (FC) concept described in CCITT Recommendation I.310 [6] than with the FE concept of CCITT Recommendation I.130 [5]. The link between the service characterisation (stage 1), the access service and the network service (stage 2) can be made using CCITT Recommendation I.130 [5] FC concept. Figure 4 shows the relationship between FC and FE, and indicates that the FC bridges across two FGs which contain FEs.



FC:	Functional Component	FE:	Functional Entity
ACC:	Access Connection Control	AC:	Access Control
CC:	Connection Control	NBC:	Network Bearer Control
NCC:	Network Connection Control	NT2:	Network Termination 2
ABC:	Access Bearer Control	LCRF:	Local Call/Connection Related Functions
TCRF:	Transit Call/Connection Related Functions		

Figure 4: Relationship between the services definition FEs and the definitions of FGs for the case of the LCRF FG

A stage 2 supplementary services description could define the access and network services for calls (monomedia and multimedia) and connection services. This description, taking into account CCITT Recommendation X.210 [11] signalling and interworking concepts and CCITT Recommendation I.310 [6], could introduce FEs: Access Control (AC); ABC (access service call components); and Call Control (CC); and NBC (network service call components).

Generic RCs could be introduced for each different service between steps 2.1 and 2.2 of stage 2 of the CCITT Recommendation I.130 [5] 3-stage procedure (also at the beginning of CCITT Recommendation Q.65 [10]).

At the beginning of stage 3 (signalling studies and protocols) a dedicated RC could be produced for each bearer and teleservice.

6.2.2 Functional entity identification

It is assumed that the FGs, as used in RCs for the control plane, are composed of FEs, AC, ABC etc. The properties of FEs are as follows:

- an FE is situated in a single location;

- FEs are network addressable entities;

- if a pair of communicating FEs are located in physically separated devices, then the information flows between them define the information transfer requirements for a signalling protocol between the devices;

- each FE is dedicated to a service being offered, e.g. AC, CC, the bearer service (AC/CC-bearer), or AC or CC for video telephony (AC/CC-videotelephony).

At this point in the development, the relationship between the FEs (a type of FG) and the FGs, such as LCRF, etc. developed for the user plane, can be seen. These two groups are now superimposed onto another multiple RC to show all three planes and their relationships, see figure 1.

6.2.3 User plane

The user plane RCs have already been described under subclause 6.1. These have been based on CEs, which support a wide range of topologies and CTs, and so are highly generic in this respect. Work is starting to relate CTs to particular CE options and instances. A generic user plane RC is shown in figure 1.

6.2.4 Network management

Network Management (NM) RCs are described within CCITT Recommendation M.3010 [7], but they have not been produced strictly starting with the concept of user services to be provided. A bottom up method has been used in which the functions needed have been taken from current Operations and Maintenance (O&M) activities and extended to include network concepts. At present, work is being undertaken to define the equivalent services (the application services) but the description of these from an NM user's point of view is more difficult than in the other planes. This is because there are two types of service, the first, dealing with deterministic, or predictable activities, and the second, which is non-deterministic, involves the continuous management corrective control loop designed to stabilise QOS/network performance.

A generic RC for NM is shown in figure 1.

Details for the construction of this RC are for further study.

7 Conclusions

Although a fully formal set of definitions and language to define and construct RCs has not been given, it is concluded that sufficient detail has been presented so that ambiguities in meaning have been reduced. The rules presented in subclause 5.5 should help to identify ambiguous RCs, particularly descriptions which are really specific realisations; and the set of definitions in Annexes A and B should help define more precisely the concepts used.

It is also concluded that the objectives set out in Clause 4 will help clarify the use of RCs.

Further work to eliminate some of the remaining problems, as described in Annex A, should also take place before a fully formal framework is in place.

To fully cover all the problems encountered in producing a formal framework might result in the need for a very formal stylised descriptive language, which may not be easy to use. In this case sets of tools will be needed to produce a user-friendly interface.

Annex A: Concepts used in this ETR

A.1 Introduction

A.1.1 The concept of "function"

This is the definition of a schema which is obeyed by instances covered by the schema. That is to say, the schema specifies unambiguously what will be the outcome of a particular function instance. Using Dom, and Ran to represent Domain and Range of input and output messages, respectively, the description of a function can proceed as follows:

> We can name any schema f.
>
> **Dom(f)** is the closure of message instances that all instances of f can accept. Informally "the possible input to f".
>
> **Input(f)** is an instance of Dom(f).
>
> **Ran(f)** is the closure of message instances that all instances of f will produce. Informally "the possible output from f".
>
> **Output(f)** is an instance of Ran(f).
>
> **State-space(f)** is the set of variables that are under control of f and the value-set permissible for each variable. It can be considered as a collection of pairs (Var:Type).
>
> State(f) is an instance of State-space (f) at a certain point in time. Different states can be denoted by decorations, like State'(f), State''(f), State(f), etc. It can be considered as a collection of pairs (Var:Value).

The function itself is defined by Map(f) which unambiguously defines the relationship between all possible pairs (Input(f),State(f)) as the first term and (Output(f),State'(f)) as the second term. The possible pairs are subsets of the set of pairs defined by ((Dom(f),State-space(f)),(Ran(f),State-space(f))). It is a subset because not all combinations of values are logically possible.

> NOTE: The definition of Map includes changes of state with or without output. It is not possible to define a Null Map, that is a Map where neither output nor a new state is the effect. The opposite is possible, that is a Map can include as effect both output and a changed state.

There are two ways to define Map(f). For simple and small Map(f) it can be enumerated. For larger and more complicated Map(f) it can be defined indirectly, for instance using Specification and Description Language (SDL). Nothing excludes the use of both approaches, for instance, some particularly important mappings can be explicitly enumerated, and thus does not affect the general mapping specification.

Trace(f) is the sequence of states from and including initial state up to and including terminal state. Once trace thus shows the history of one function instance.

Trace-space(f) is the set of all possible traces of f.

A.1.2 The concept of "atomic function"

An atomic function is a function that is not decomposable. That is to say, it is not possible to decompose any of its characteristics: the domain, the range, the state-space and the mapping. The internal state of an instance of an atomic function is not observable. The atomic function can only inform about its state via messages that are part of its range. From this, it follows that the execution of an instance of an atomic function cannot be controlled during the execution. The only way to control it is via messages that are part of its domain (=parameters).

A.1.3 The concept of "elementary function"

A function allocated physically into one limited physical unit. This physical unit is defined as architectural entity which can be similar to a self-contained computer. It is recognised that a reallocation of physical units may have the effect that elementary functions can be allocated to two or more physical units and thus become global functions, CCITT Recommendation I.130 [5].

A.1.4 The concept of "functional group"

There are at least two ways to group functions, in "series" or in "parallel". One (serial) is the concatenation of two functions f and g such that:

Ran(f) = Dom(g). From the outside it means that the interface between f and g disappears. This group is here called function concatenation.

The other way (parallel) is to group function is to join them. This means that if two functions f and g are joined into h, Dom(h) = Dom(f) U Dom(g), Ran(h) = Ran(f) U Ran(g), State-space(h) = State-space(f) U State-space(g) and Map(h) = Map(f) U Map(g). It follows that the joined mapping must be consistent. The reason for joining functions is that they are related and should be treated as one logical unit. Most often the joining will be accompanied by the introduction of a function that controls the execution of the joined functions (e.g. a failure handling module where the control of failure processing is based on failure codes.)

This kind of group is here called function join.

The activation of a FG does not necessarily mean that all components are activated at the same time. Neither is it true that the components of a FG terminate at the same time. The activation and termination of functions belong to the executing semantics of the function, which is not related to its conceptual definition.

"Function group" is a synonym for "functional group".

A.1.5 The concept of "global function"

A global function is a function that physically is distributed over two or more physical units, which are defined as architectural entities which are similar to self-contained computers. It follows that a physical reallocation of functions where a global function is allocated to only one physical unit makes the function elementary, CCITT Recommendation I.130 [5].

A.1.6 The concept of "functional entity"

A functional entity is a function (atomic or group) where the component functions together are necessary and sufficient to perform a given task, (CCITT Recommendation I.310 [6]). From CCITT Recommendation Q.65 [10] it follows that a FE is a group of functions that is put in a single location. Informally it can be considered as a machine specialised for a particular task, which is needed to execute services.

There is no name for the negation of "functional entity", which is the same as a FG where the components are not necessary and sufficient for a particular task. They can have been grouped for other reasons. If follows by logic that an atomic function always is a FE.

A.1.7 The concept of "functional element"

This must be the same "function", and can, therefore, be either "atomic function" or "functional group".

A.1.8 The concept of "functional component"

A set of elementary functions performed in an order that yields a specified result, CCITT Recommendation I.310 [6]. A FC always has an invoking and a responding entity. The invoking entity is the entity which acts in response to an FC request from an invoking entity. Therefore, during its lifetime, the FC forms a particular kind of FG.

A.1.9 The concept of "elementary messages"

This is here defined to be "a collection of semantically referenced data terms that are necessary and sufficient to convey an unambiguous piece of information".

The fact that the data terms are necessary means that an instance of an elementary message must not rely on a context to be able to be interpreted unambiguously. All references must be there.

The fact that the data terms are sufficient means that an instance of an elementary message forwards one single piece of information, not more.

Joined messages that cannot be interpreted without guide for how references should be treated are here called "message joins". They are the same as "records" in computer terminology.

NOTE: An elementary message is also a record.

Elementary messages and message joins are both called "messages". This holds recursively, so a join of message joins is also a message join and is called "message".

A.1.10 The concept of "message type"

This is a schema for a message, where there is a data type definition in place for each data term. Therefore the closure of a message type is the cartesian product of the component types.

A.1.11 The concept of "reference point"

This is a very difficult point. Here we discriminate between Context Free Reference Point (CFRP) and Context Dependent Reference Point (CDRP).

The maximal CFRP between two functions f and g is defined as Dom(f) U Dom(g) U Ran(f) U Ran(g). This is true for a two-way connection between f and g. The Dom and Ran are defined in terms of messages, preferably elementary messages.

A CFRP is said to be context-free because the specification does not take into account that the actual acceptability of a particular message instance depends on the actual state of the receiving function instance.

An CDRP is said to be context-depending because the acceptability of message instances is depending on the actual state of the receiving function instance. Therefore, it follows that a description of a CDRP must include the state of the receiving function instance. This may lead to very large specifications.

By convention a RP is considered to be a CDRP. From this follows that the description of a RP must include the states as was stated above.

A.1.12 The concept of "message flow"

This is a synonym for CFRP and also for "data flow".

A.1.13 The concept of "information flow"

It has been found that information is the interpretation from data using the conventions for interpretation of the data terms in a message, and their semantic role in the message.

This concept, "information flow" cannot exist because information which is not interpreted cannot flow. The proper name for the intended concept is "data flow" or "message flow" or "CFRP".

Where it exists in CCITT Recommendations (e.g. CCITT Recommendations I.130 [5] and I.310 [6]) it should be considered as a synonym for "data flow".

A.1.14 The concept of "information attributes"

This concept is not possible in the context, because it means the attributes of the interpretations that are made from messages. What is meant is probably exactly what here is called CFRP, that is a set of possible messages. The proper name for the concept is "data flow attributes".

A.1.15 The concept of "information flow attributes"

It has been shown that the proper names are "data flow attributes" or "message flow attributes" or "CFRP attributes".

If the CFRP is considered as an object, it can have properties or attributes related to it. These attributes are related to the CFRP itself, but a closer analysis may show that the attributes could better be associated with the neighbouring functions.

This allocation of attributes is for further study.

A.1.16 The concept of "object"

Anything in an existing system that can be said to have an individual existence can be said to be an object. Objects of a similar kind are defined as object class. The specification of an object class normally includes its data space and rule space. Each time an individual object is created, its data space and rule space is created. In an implemented and executing system, each object will have a physical location. It can be distributed, although this causes implementation problems. Distribution (like that for "global functions") can be better realised by means of references to other objects (=local functions).

A.1.17 The concept of "identity"

In general, x is identical to y if they are clones. This means that when they are executed, their state sequences (=traces) will be identical. From this follows for two functions f and g: Dom(f) = Dom(g) & Ran(f) = Ran(g) & State-space(f) = State-space(g) & Map(f) = Map(g) & Trace-space(f) = Trace-space(g). The real-time properties may differ if they are implemented on different machines.

A.1.18 The concept of equivalence

This can also be called observational equivalence. This means that when two functions f and g execute with identical input sequences, their output sequences will be identical. However, it is not necessary that state-space or trace-space are identical. From this follows that the mapping is not identical, because the internal state changes may be different.

This will be true for two functions f and g: Dom(f) = Dom(g) & Ran(f) = Ran(g) & Map(f) = Map(g) with respect to input $\rightarrow$ output, but not with respect to State $\rightarrow$ State'.

A.2 The orthogonal relationship between types of functions

We are separating completely descriptions on the conceptual level from the physical level.

On the conceptual level, functions are either atomic or FG. On the physical level, functions are either elementary or global. These distinctions are orthogonal, meaning that the following combinations are possible:

1) conceptual atomic; physical elementary (one function in one physical unit);

2) conceptual atomic; physical global (one function distributed over at least two physical units);

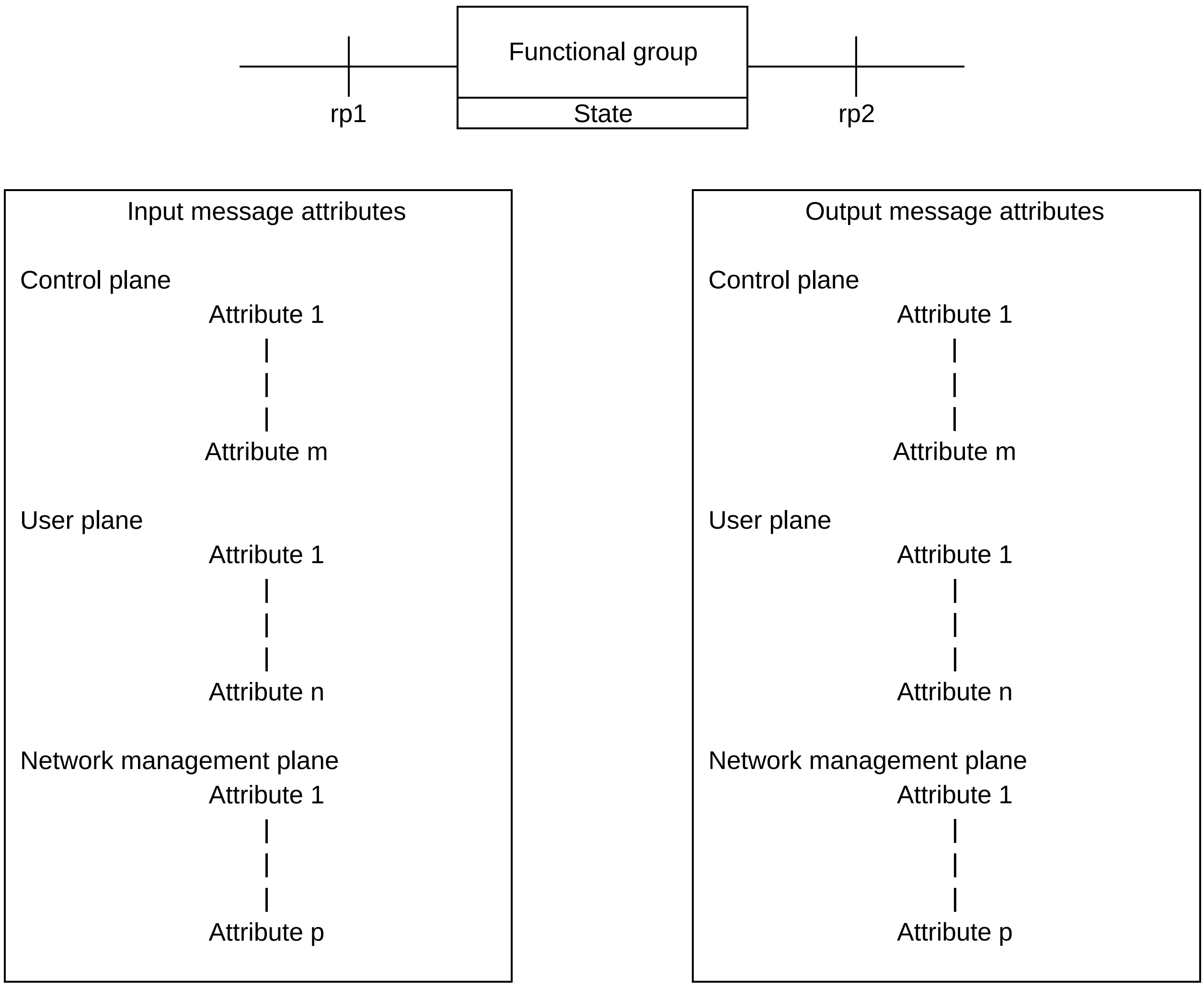
3) conceptual FG; physical elementary (at least two functions forming a group and allocated in one physical unit);

4) conceptual FG; physical global (at least two functions allocated over at least two physical units).

NOTE: The property of a function to be a FE is a property in the conceptual domain and has nothing to do with physical allocation. However, see subclause A.1.6 and CCITT Recommendation Q.65 [10].

Annex B: Specification of functions and messages

A template for the specification of a FG and RPs is given in figure B.1. The FG specification involves descriptions of the attributes of the incoming and outgoing data or messages. At a generic level, these will comprise brief statements of attributes as shown in figure B.1. A full description can later be made when details of attribute values and message formats are known. The attributes may contain CT attributes associated with the functions within the FGs.



rp: reference point.

Figure B.1: Template for FG and RP definition

Where a FG is involved, it will usually be partitioned into separate functions. In this case, the associated message or data attributes may be similarly partitioned.

History

Document history	
February 1993	First Edition
January 1996	Converted into Adobe Acrobat Portable Document Format (PDF)