INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

I.328/Q.1202

(09/97)

SERIES I: INTEGRATED SERVICES DIGITAL NETWORK

Overall network aspects and functions – Reference models

SERIES Q: SWITCHING AND SIGNALLING

Intelligent Network

Intelligent Network – Service plane architecture

ITU-T Recommendation I.328/Q.1202

(Previously CCITT Recommendation)

ITU-T I-SERIES RECOMMENDATIONS

INTEGRATED SERVICES DIGITAL NETWORK

GENERAL STRUCTURE	
Terminology	I.110–I.119
Description of ISDNs	I.120–I.129
General modelling methods	I.130–I.139
Telecommunication network and service attributes	I.140–I.149
General description of asynchronous transfer mode	I.150–I.199
SERVICE CAPABILITIES	
Scope	I.200–I.209
General aspects of services in ISDN	I.210–I.219
Common aspects of services in the ISDN	I.220–I.229
Bearer services supported by an ISDN	I.230–I.239
Teleservices supported by an ISDN	I.240–I.249
Supplementary services in ISDN	I.250–I.299
OVERALL NETWORK ASPECTS AND FUNCTIONS	
Network functional principles	I.310–I.319
Reference models	**I.320–I.329**
Numbering, addressing and routing	I.330–I.339
Connection types	I.340–I.349
Performance objectives	I.350–I.359
Protocol layer requirements	I.360–I.369
General network requirements and functions	I.370–I.399
ISDN USER-NETWORK INTERFACES	
Application of I-series Recommendations to ISDN user-network interfaces	I.420–I.429
Layer 1 Recommendations	I.430–I.439
Layer 2 Recommendations	I.440–I.449
Layer 3 Recommendations	I.450–I.459
Multiplexing, rate adaption and support of existing interfaces	I.460–I.469
Aspects of ISDN affecting terminal requirements	I.470–I.499
INTERNETWORK INTERFACES	I.500–I.599
MAINTENANCE PRINCIPLES	I.600–I.699
B-ISDN EQUIPMENT ASPECTS	
ATM equipment	I.730–I.749
Management of ATM equipment	I.750–I.799

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

ITU-T Q-SERIES RECOMMENDATIONS

SWITCHING AND SIGNALLING

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

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

ITU-T RECOMMENDATION I.328/Q.1202

INTELLIGENT NETWORK – SERVICE PLANE ARCHITECTURE

Summary

This Recommendation has been updated to include new capabilities and functionalities included in the IN CS-2 series (Q.122x) of Recommendations. It provides a general architecture of the IN service plane in a way that specific functionalities and their interactions can be identified and described in other Recommendations which make reference to the service plane. The service plane illustrates that IN-supported services can be described to the end user or subscriber by means of a set of generic building blocks called "service features". This Recommendation describes this process of supporting services in an IN-structured network and the service plane's relationship to the other planes in the IN Conceptual Model.

Source

ITU-T Recommendation I.328/Q.1202 was revised by ITU-T Study Group 11 (1997-2000) and was approved under the WTSC Resolution No. 1 procedure on the 12th of September 1997.

FOREWORD

ITU (International Telecommunication Union) is the United Nations Specialized Agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the ITU. The ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 1.

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

NOTE

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

INTELLECTUAL PROPERTY RIGHTS

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

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

CONTENTS

INTELLIGENT NETWORK – SERVICE PLANE ARCHITECTURE

(revised in 1997)

1 General

Recommendation I.312/Q.1201 "Principles of intelligent network architecture" presents the Intelligent Network Conceptual Model (INCM), as based on a four-plane structure.

The objective of this Recommendation is to provide a general architecture of the IN service plane in a way that specific functionalities and their interactions can be identified and described in other Recommendations making reference to the IN service plane architecture contained in this Recommendation.

2 Service plane architecture

2.0 General

The service plane illustrates that IN-supported services can be described to the end user or subscriber by means of a set of generic blocks called "service features".

A service is a stand-alone commercial offering, characterized by one or more core service features, and can be optionally enhanced by other service features.

A service feature is a specific aspect of a service that can also be used in conjunction with other services/service features as part of commercial offering. It is either a core part of a service or an optional part offered as an enhancement to a service.

The service plane represents an exclusively service-oriented view. This view contains no information whatsoever regarding the implementation of the services in the network (for instance, an IN type of implementation is invisible). All that is perceived is the network's service-related behaviour as seen, for example, by a service user.

Furthermore, management services are contained in the service plane; they can be described to the end user by means of service management features.

2.1 Characterization of services and service capability requirements

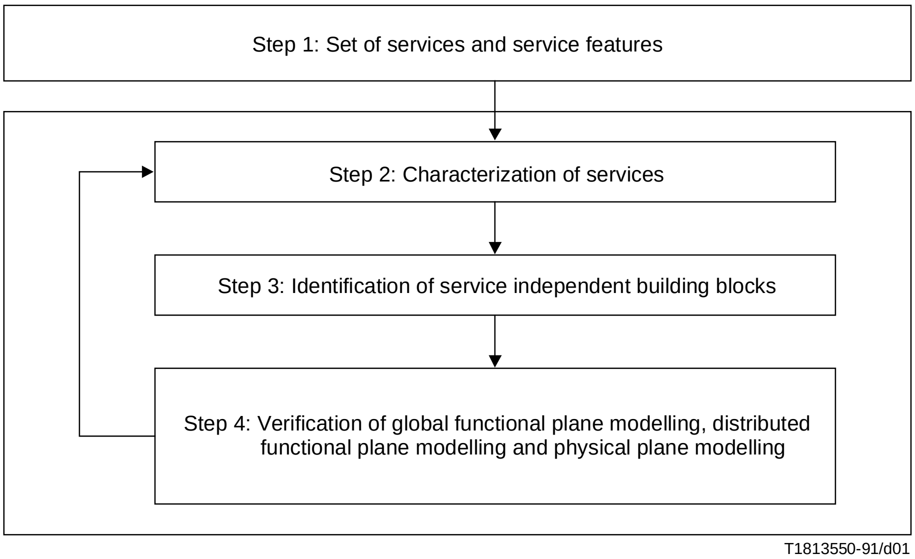
Characterization of services and service features is to identify service independent capabilities that are required to construct and/or customize services by the users or network operators. Examples of service capabilities required from the user point of view are call queueing, customized announcement, etc.

There is a need for a structured approach with which to classify service characteristics and identify service capabilities. The structured approach shown in Figure 1 below demonstrates a high level approach for analizing services and decomposing services into Service Independent Building Blocks (SIBs) These reusable service independent building blocks (such as TRANSLATE, USER INTERACTION or CHARGE) will form the basis for input to global functional plane modelling and distributed functional plane modelling.

It is recommended that activities involving functional modelling make use of the results of such service analysis, based on the characterization of services for verification of their models, and to ensure a unified model for service processing.

2.2 Service plane modelling

Services are comprised of one or more Service Features (SF). A service feature is the smallest part of a service that can be perceived by the service user. These SFs can also be used as building blocks in the specification and design of new, more complex services. SFs are comprised of one or more SIBs which are described in Recommendation I.329/Q.1203.



Step 1 Existing service descriptions (Stage 1, Service description) as well as emerging service descriptions are selected as candidates for analysis.

Step 2 These services are characterized based on the principle of decomposition of services into functions.

Step 3 The analysis in step 2 results in requirements in the form of service independent building blocks as the input to global functional plane modelling, distributed functional plane modelling and physical plane modelling.

Step 4 Verification of the global functional plane modelling, distributed functional plane modelling and physical plane modelling results in improved SIBs through feedback to step 2.

Figure 1/Q.1202 – An approach for analyzing services

All individual telecommunication services identified in the service plane should be described as seen from the user's viewpoint without reference to how the services are implemented in the network.

In the service plane architecture, it is stressed that all capabilities experienced by a service user of the network represent telecommunication services (basic or supplementary). The service user may make use of the service for his own communication needs or may combine a number of services together and with perhaps additional capabilities, use the combination as a means of providing communications to other (third) parties.

2.3 Service and service feature interaction

This subclause focuses on the interaction between IN supported services and other supplementary services and not between basic services. The service interactions are described from the customer and user point of view.

There are often many ways to deal with an interaction between two or more services. In an IN-structured network, service interactions may be customized.

Example of service interactions are given below:

- Abbreviated Dialling and Number Screening;
- Freephone and Call Forwarding Unconditional;
- CLIR (Calling Line Identification Restriction) and CLIP (Calling Line Identification Presentation);
- Call Forwarding and Premium Rate Service;
- Call Waiting and Call Forwarding Busy;
- Conference Call and CUG (Closed User Group);
- Meet-me Conference and CUG.

Examples of different ways to treat interactions between Freephone and Call Forwarding Unconditional are:

1) Freephone call attempts shall be forwarded like other terminating calls.

2) A Feephone destination shall be selected for Freephone calls, even if it has activated Call Forwarding Unconditional.

3) A Freephone destination shall not be selected for Freephone calls if it has activated Call Forwarding Unconditional.

2.3.1 The service and feature interaction problem

NOTE – In the following text, in order to facilitate the reading, the term "service" is currently used instead of "service feature".

2.3.1.1 General introduction

This subclause provides a definition of the interaction problem and a setting of the scope of interaction studies.

IN-structured networks are used to provide rapid deployment of new services and service features to telecommunications users. These services and service features are mostly specified by different designers and implemented on different networks. This increases problems occurring during the life cycle of each service because the resources (network resources, services data) are influenced by other services and service features and thus such a service behaves differently.

For the services to behave according to their specification and to fulfil the constraints imposed by sharing of resources among different services, methods for the detection and resolution of interaction situations must be found.

It is to be noted that a service is supported within a *service addressing domain*. Consequently, interaction can only be addressed (and solved) within such a given addressing domain. Interaction between two network-specific services, e.g. interaction between a call rerouting in a PBX and a service in the public network, cannot be solved as far as the services are not to be considered as multi-network, i.e. applied to a multi-network addressing domain.

Service interaction applies to all interactions of the service being described with other services which have been already identified.

Service feature interactions may occur (for example):

1) among different features associated with the same service;

2) between features associated with a service for a given service user and features associated with other services the same user may have requested or been assigned;

3) between features associated with a service for a given service user and features associated with possible services related to the terminal/calling line that the user is currently using, e.g. in case terminal mobility and/or personal mobility are involved.

An IN-structured network may handle multiple services for the same call. The necessary interactions shall be defined for the processing of several services for the same call. When multiple services can be activated concurrently, some prioritization of services will be necessary. User specific requests may take priority over a group service request. Additionally, certain services may override or deactivate other services.

The service interaction is part of the specification of services, and should be dealt with in the service plane modelling.

How service interactions are implemented is not visible in the service plane. The usage of the service independent mechanism in the IN architecture to handle service interactions will be visible in the global functional, the distributed functional plane and the physical planes.

The following issues need consideration when service interactions are specified:

– different phases of a call, i.e. originating, terminating, interrupt (active) and release phase of call processing;

– a service spans more than one network. This may impose additional requirements on service interaction, which is for further study;

– service interaction may occur between services offered to a single user, as well as between services offered to different interacting users.

2.3.1.2 Overview

This subclause, after a presentation of general statements and examples, proposes a view of the service and feature interaction problem from the service plane, which tries to answer three questions that occur when considering service interaction:

– A solution: Which one? (i.e. the solution is not unique, and a choice is to be made);

– A solution: When? (i.e. in which moment of the interaction life cycle the problem is to be solved);

– A solution: How? (i.e. by which means can the problem be discovered and solutions be found).

An approach combining three mechanisms is proposed:

– spotting during service creation;

– implementing a management arbitrator;

– implementing an execution arbitrator.

This last mechanism implies a set of solutions impacting the Distributed Functional Plane, solutions built around execution managers and improved signalling, which are described in Recommendation Q.1204.

2.3.1.3 Situations of interworking between services

2.3.1.3.1 Two services involved

Considering the introduction of a new service in the network, when this new service interworks with an already existing one, three situations can be distinguished:

– when the new service interworks with an existing service without one influencing the other, its introduction has *no impact*, from an interaction point of view, on the considered service: the two services are functionally independent;

– when the aim of the new service is to modify an existing service, there is a *desired interworking* (e.g. Freephone modifies the charging of the basic telephony service): the two services are functionally complementary;

– when an undesired consequence of the mutual influence of the new service with already existing services is observed, such an *undesired interworking* is called *interaction*. In this case, services may be either functionally complementary (some precautions have to be taken to get a correct interworking), or functionally incompatible (the second request shall be refused or shall override the earlier one).

One may notice that service interaction may be **technical** when interworking situation results either in the dysfunction of one of the services involved or in the immediate subsequent dysfunction of another service (in such a case, one of the services does *not* fulfil its technical basic requirements), or **policy** when the effect is unacceptable to the service user or the service subscriber from an economical, sociological or ergonomical point of view, or is deemed as unacceptable to business by the service provider or the network operator (such a situation occurs *even if* each of the services does fulfil its technical basic requirements).

2.3.1.3.2 More than two services involved

Multiple interactions, i.e. interactions involving more than two services, may be divided into three cases:

– bilateral interaction (i.e. considering S1-S2, S2-S3 and S3-S1 independent interactions);

– multilateral interaction (i.e. considering global S1-S2-S3 interaction);

– roundabout interaction (i.e. considering the case where S1-S2 interaction causes an S1-S3 or S2-S3 interaction).

This Recommendation does not consider specific requirements for addressing such multiple interactions.

2.3.1.3.3 A thesaurus of examples

Basic examples referred to below may be found in Annex B. These are the following:

– Call Forwarding Unconditional (CFU) and Terminating Key Code Screening (TKCS).

– First case: A user forwards his calls to a line screened by a key code.

– Second case: A user has his line screened by a key code and makes his calls forwarded.

– Call Forwarding Unconditional and Automatic Call Back (ACB).

– First case: User B has requested a CFU to C, user A places a call to B which is forwarded to C, who uses Automatic Call Back).

– Second case: User B has requested a CFU to C, B calls A, and A tries to use Automatic Call Back.

– Call Forwarding Unconditional and Terminating Call Screening (TCS): Interworking between these two services presents 36 different cases to scan.

2.3.2 A solution: Which one?

2.3.2.1 A non-deterministic problem

If an end user sets both a Call Forwarding on Busy (CFB) service and a Call Waiting (CW) service on the same line, only one service can operate when a call is received in a busy situation. Such an easy example of undesired interaction seems easy to manage, and a number of solutions spring to the mind: one could forbid CFB and CW to be simultaneously active on the same line, or one could specify that CFB is to be invoked only if CW can not be invoked (for instance, if the user's line is fully busy, i.e. when the user is already involved in two calls), or one could even imagine to alter CFB, which could be activated only if the user does not accept the CW and hence would become Call Forwarding on No Reply.

This example shows that interaction may proceed from a choice. Another common example, interaction between Freephone and Call Forwarding Unconditional, shows that service interaction is not to be considered as a technical problem, and that solving an interworking problem is not deterministic at all; this interaction may be handled in three different ways, independently of any technical consideration:

– Freephone call attempts shall be forwarded like other terminating calls.

– A Freephone destination shall *not* be selected for Freephone calls, if it has activated Call Forwarding Unconditional.

– A Freephone destination shall be selected for Freephone calls, *even* if it has activated Call Forwarding Unconditional.

2.3.2.2 Goals of a feature

Considering example of CFU and ACB in B.3.2, it may be noticed that the choice of a solution depends on the goal the service user has when activating CFU. Such a user may want to forward his calls either because he wants to complete to the forwarded-to network address, or because he wants not to be disturbed by any phone call.

The coexistence of several possible goals for a given feature makes the treatment of a policy interaction more complex, for what is acceptable for one user may be unpleasant for another one.

2.3.2.3 Service provider's choices

The treatment of an interaction case has to be considered in two different steps:

– As a first step, during the specification phase, the service provider's technical staff analyzes the interworking between the two services and deducts the interaction cases which may be a problem for any of the users.

– As a second step, both technical and commercial staffs make choices amongst the different possible solutions, taking into account:
  - technical possibilities such as network capabilities;
  - ergonomical constraints (which solution will be the easiest for the users?);
  - sociological data (which solution will be the most admissible for the users?);
  - economic factors (does the perfect ergonomical solution bring an excessive extra cost?);
  - user’s requirements.

However, the service provider’s choice may also be to let the user himself decide which way the two services will complete together, i.e. to leave to the final user the choice of the interaction solutions which have been brought up.

2.3.2.4 Customer’s choices

According to the pair of interacting services, the customer’s choice may intervene in different states of the service life cycle.

In the case of interaction between CFU and ACB (Figure B.4), the user may choose between three solutions when he subscribes to the service. In the case of interaction between CW and CFB, the user may choose at activation time if he prefers to give the priority to the CW with CFB only in case of full busy, or to give the priority to the CFB.

Another example is given in the case of a user having set up his subscription profile such that he has a wake-up call at 0530 a.m. each working day. One day, he activates a CFU to a friend. What happens when the morning wake-up call arrives? One can consider that probably the CFU is activated because the user accidentally forgets to deactivate it after having spend a couple of hours at a friend’s place, thus the CFU has to be overridden. But one can also think that maybe the user is still staying at a friend’s place (e.g. watching the cat while the friend is away) and actually wants his wake-up call forwarded. Typically, that kind of interaction choice may be included in the subscriber’s profile.

2.3.3 A solution: When?

2.3.3.1 Concept of interaction life cycle

An interaction is characterized by an unexpected and somewhat defective service demeanour, whereas service design shall always result in a healthy system behaviour.

When investigating the reasons of such a defective demeanour, it appears that, for several cases mentioned in Annex B, an interaction between two services is characterized by an execution problem generated by a *previous data modification*. In every case, the activation of a service (in Annex B, CFU) is actually at the origin of the interaction:

– in Figure B.1, from user A’s point of view, because of the activation of CFU, there is a dysfunction when activating TKCS;

– in Figure B.4, from user C’s point of view, because of the activation of CFU, there is a dysfunction when invoking ACB;

– in Figure B.5, from user A’s point of view, there is an annoyance (not really a dysfunction) because of the activation of CFU.

This remark leads to introduce the concept of interaction life cycle.

2.3.3.2 Interaction "wild life cycle"

As a first step, consider interaction between two conflicting services left to itself, without any kind of intervention of the service provider, in the frame of its so-called "wild life cycle" (see Figure 2).

2.3.3.2.1 Interaction germination

Interaction germination is defined as a data[1] modification or initialization which prepares an *interaction manifestation* that may occur either later on in the same call or in a further call, implying risks of dysfunction.

---

[1] The notion of data is to be considered in a broad sense, to integrate the concept of resource.

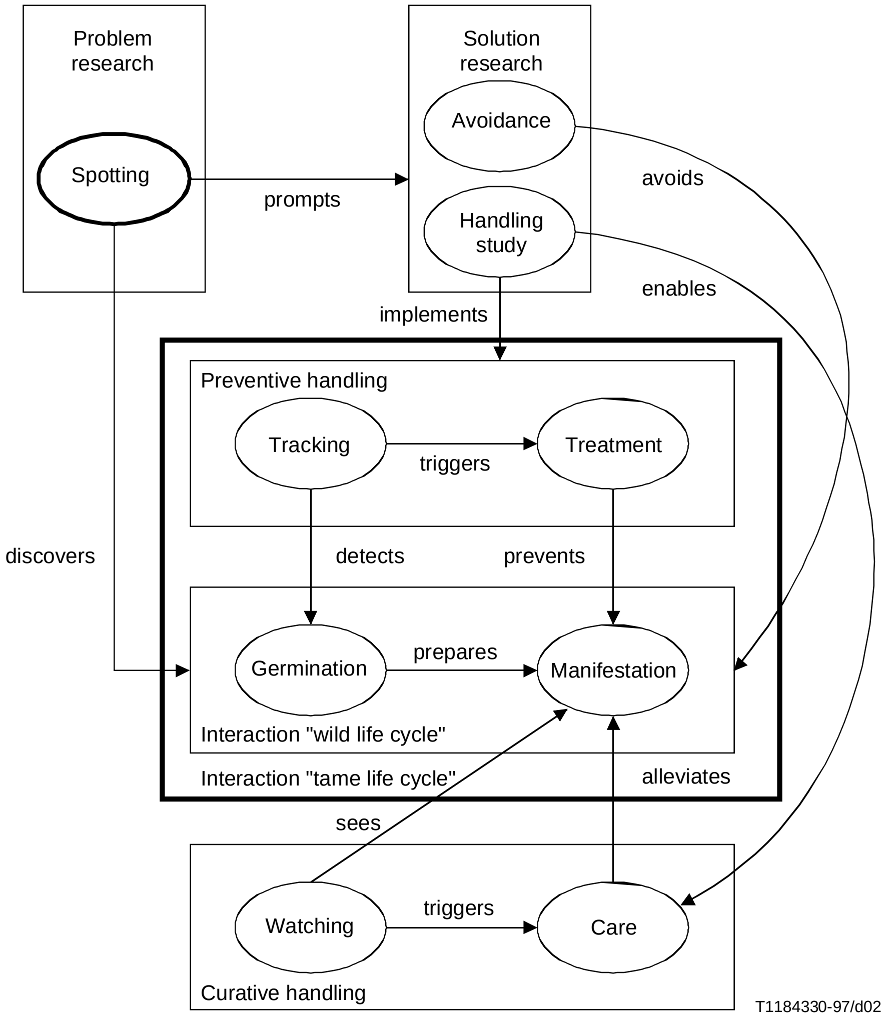


Figure 2/Q.1202 – The interaction life cycle

Interaction germination can take place either at service management (data modification or service activation) or at service invocation.

2.3.3.2.2 Interaction manifestation

Interaction manifestation, defined as the moment when an interworking between services causes a situation viewed as unsatisfactory from any of the users, occurs during service invocation. It happens:

– either when a service disturbs or inhibits the expected execution of another service considered separately (or of another instance of the same service);

– or when the joint accurate executions of services creates new unforeseen behaviours which would not appear in case of executions considered separately.

2.3.3.3 Interaction research

Interaction research takes place during service creation, and is therefore performed in the Service Creation Environment (SCE). It is made of three parts:

– **spotting** aims at finding as many interaction cases as possible, by analyzing the interworking between the service to be introduced and the existing ones;

– **handling study** aims at choosing, for each case, a solution among the possible ones (according to network capabilities, ergonomical constraints, sociological data, economical factors and user's requirements);

– **avoidance** aims at decreasing the number of cases to be handled.

2.3.3.3.1 Interaction spotting

Interaction spotting takes place during the various phases of service creation. It consists in the analysis of the new service model, in conjunction with already existing service models, in order to find as many interaction cases as possible.

One of the basic aims of *interaction spotting* is the discovery of potential *interaction manifestations*, as well as *interaction germinations* which prepare them, in order to facilitate subsequent *interaction handling*.

Note that we use the term **interaction spotting**, to characterize the moment when an interaction is foreseen at service *creation*, in order to make a distinction with "detection" at service *execution*, for which we shall propose other wordings. In technical literature, the term **detection** is often used in both cases.

2.3.3.3.2 Interaction handling study

Interaction handling study is performed by both service provider's marketing and technical staffs. It evaluates the gravity of each spotted case and, for those that can not be solved by avoidance techniques, deduces handling mechanisms necessary to their detection and treatment:

– *late (*or *curative) handling* watches interaction manifestation and tries to alleviate its effects;

– better, *early (*or *preventive) handling* intends to forbid germination by tracking harmful data configurations.

2.3.3.3.3 Interaction avoidance

Interaction avoidance takes place during service creation. It aims at eliminating, once and for all, some interaction cases; thus, it can be compared to prophylaxis methods. We consider as interaction avoidance the development of more advanced terminals, the introduction of new signalling capabilities in the network, as well as the redefinition or restriction of service functionalities, or the determination of acceptable service interworking.

2.3.3.4 Curative handling

Curative handling, acting at execution level inside the Service Control Environment (SCE), watches and remedies to an *interaction manifestation* already visible.

2.3.3.4.1 Interaction watching

Interaction watching is the observation of service execution, in order to ensure that no *interaction manifestation* occurs and, in the opposite case, to trigger mechanisms designed to take care of this manifestation.

2.3.3.4.2 Interaction care

Interaction care includes the processing of any mechanism designed to solve an *interaction manifestation* after its detection by *interaction watching*. Since the manifestation is already visible by one of the users, it can not be avoided, and its effects can only be alleviated. For instance, the calling user may be prompted to take a decision related to the completion of his call.

2.3.3.5 Preventive handling

Preventive handling intends to avoid any *interaction manifestation* by implementing mechanisms tracking *interaction germination* as soon as it occurs. The interaction process is then totally controlled and interaction performs its so-called "tame life cycle". Such handling also takes place inside the Service Creation Environment (SCE).

2.3.3.5.1 Interaction tracking

Interaction tracking is defined as the moment when an *interaction germination* can be observed during service management (data modification or service activation) before any *interaction manifestation* occurs. It allows arming of mechanisms which will enable *interaction* treatment.

2.3.3.5.2 Interaction treatment

Interaction treatment is the processing of complementary mechanisms designed to solve an unsatisfactory interworking situation, which may have germinated either in the same call or in a previous call.

Such a processing is a consequence of *interaction tracking.* However, it may take place either **before** (if it is preventive), **while** or **after** (if it is curative) *interaction germination.* Refer to the examples:

– in cases 19 to 24 in B.4 (case *before*), if B wants to activate a CFU to C while C has entered B in his blacklist, these services will include incompatible data, and a clever management is to forbid the data modification that creates this incompatibility;

– in Figure B.2 (case *while*): a possible solution is to take into account a predetermined invocation order;

– in Figure B.4 (case *after*): a possible solution to the conflict is to not take into account the second service.

2.3.3.6 Semantic precision

The word "detection" is not used to name the phases of the interaction life cycle, for it is a generic term usually turned to fit any case, from the moment when an interaction is *foreseen* at service *creation* to the moment when its *manifestation* is noted at service *execution*.

If "detection" may be kept as a generic name (though it implies some ambiguities), three distinct activities in interaction resolution are christened after three terms keeping the semantic peculiarities of common English language:

– **spotting**, which implies an idea of location of targets and an idea of investigation, seems perfectly applicable the active task to be performed during service creation;

– **tracking**, which implies an idea of following a (moving) target, fits the active task of detecting germination at service execution;

– **watching**, which implies an idea of keeping under observation and an idea of continuous attention, suits the passive task of noticing interaction manifestation after it occurred.

2.3.4 A solution: How?

2.3.4.1 Towards an integrated solution

This discussion uses a comparison to the interaction life cycle and a disease life cycle.

– As a first step, the process "contamination illness" is discovered by medical research. By analogy, the "germination manifestation" process, so-called "wild life cycle", is spotted. The **spotting** result includes the description of each interaction manifestation, but also the characteristics of the associated germination.

This discovery is followed by a phase in which decisions of disease handling have to be made, according to the results of pharmaceutical research. This corresponds to the **study of interaction handling** in which choices are made, taking into account technical, sociological, ergonomical and economical constraints. The solution of an interaction case never proceeds from determinism, and the service provider may choose among open possibilities, which may be implemented via three techniques.

– Prophylaxis propositions intend to suppress the contamination by appropriate means, such as vaccination or impeding transmission of the virus. This corresponds to the so-called **avoidance techniques**, implemented in the Service Creation Environment (SCE): service specifications refinements, underlying network adaptations and signalling systems enhancements are proposed to impede germination. Thus interaction "wild life cycle" is suppressed.

– When prophylaxis is insufficient, one may pin all hopes on a curative treatment to alleviate illness manifestations. This corresponds to the implementation of an **execution arbitrator** which has, during service execution and according to the choices previously made, to watch and deal with interaction manifestation. It is implemented in the Service Execution Function within the Service Control Environment (SCE).

– When pharmaceutical research offers a preventive treatment, methods to detect contamination are implemented and the treatment is triggered to prevent illness. This corresponds to the implementation of a **management arbitrator**, a data manager able to track germination and to treat it according to the choices previously made. It is implemented in the Service Execution Function within the Service Control Environment (SCE).

None of these methods is self-sufficient, and only a mix may get through the whole set of interaction problems, in order to tackle the service interaction problem as early as possible in both the SCE and SEF within the SCE by merging different compatible approaches.

NOTE – Spotting mechanisms are sometimes referred to as "Static mechanisms", and techniques related to the execution arbitrator as "Dynamic mechanisms".

2.3.4.2 Spotting in the SCE

Since spotting is a prerequisite for avoidance or handling, the investigation shall primarily consist in spotting all potential interaction manifestations and germinations. Spotting is a complex activity, due to:

– the increasing number of new services;

– the intrinsic complexity of many services and of the different ways in which services can interact;

– the fact that interaction can occur between more than two services;

– ergonomical and sociological aspects involved by the notion of unwanted or unforeseen behaviours.

Some possible spotting techniques are mentioned in Annex B.

2.3.4.3 The resolution techniques

2.3.4.3.1 Avoiding interaction

After spotting activity has been performed, one has to investigate if the interaction problem can be avoided, applying the techniques listed above (service specifications refinements, underlying network adaptations and signalling systems enhancements). One shall take into account the various users' interests, looking for the best compromise between them.

Note that when an interaction is avoided during the creation process, the decision taken and the causes of such a decision shall be carefully recorded. A future evolution of either the network or the services might involve reconsidering restrictions brought to service functionalities.

2.3.4.3.2 Tracking interaction germination

Interaction resolution may be confided to a manager (machine, functional entity) which intervenes at service execution (see Figure 3a). In such a case, interaction is solved when its manifestation occurs, with the traditional scheme: some causes generate effects, thus it is necessary to repair damages.

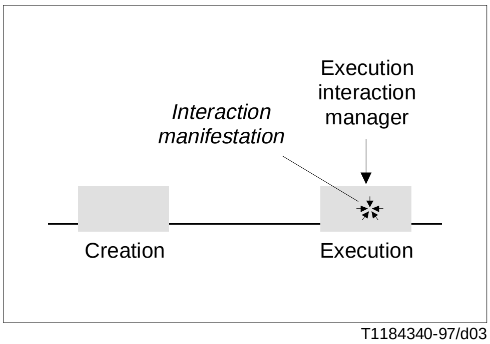


Figure 3a/Q.1202 – Interaction process without management at germination

But if interaction is managed as soon as germination is detected, i.e. a specific management arbitrator is invoked at each time data are created or modified, the need of intervention at the call processing level is minimized (see Figure 3b). This process looks more mature: having detected the causes before possible problems, there is no effect (it could be compared to a treatment applied after a systematic virus screening, and efficient before any symptom is visible). It restricts as far as possible the need of intervention at call processing level.

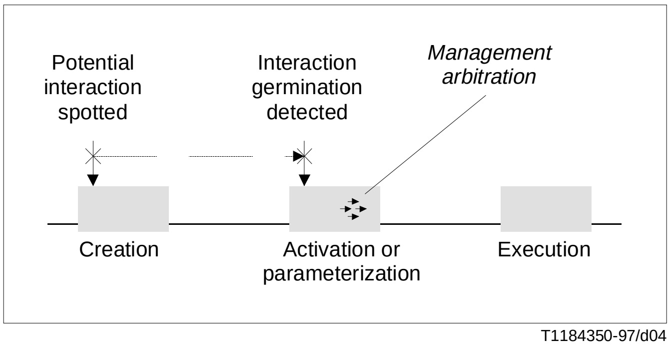


Figure 3b/Q.1202 – Interaction process with management at germination

Such a prevention can be performed by detecting germination, thus allowing the tracking entity to trigger intervention of a management arbitrator, designed to solve interaction cases during service registration or activation, by preservation of data consistency either inside a user's profile or between several user's profiles. A leading role may be devoted to the user, who could have to take a decision related to the current service context.

For the user as for the service provider, consequences are positive:

– The user is not informed of a difficulty when he places or receives a call, but when he modifies his profile, i.e. at a moment when he is generally more free-minded.

– The system reaction time can be longer in a management phase than in a call processing phase, for real time constraints are not the same, and this simplifies the service provider's task.

Applying this principle to the pairs of services taken in the example above:

– for the case in Figure B.1, the CFU user may be asked, when activating the service, to give the forwarded-to line's key code: the germination is definitely detected as soon as the activation, i.e. during user's management;

– for the case in Figure B.4, germination may be anticipated by asking the user to choose the interaction behaviour either at provision or at activation.

Some feedback is to be given to the SCE, in order to provide information about the problems met at execution.

2.3.4.3.3 Watching interaction manifestation

But there is a limit to interaction management at germination: in Figure B.5, no detection is possible in a management phase (nothing could indicate that, in a future call, a link between users A and B will exist).

Furthermore, performing though the tools available in the SCE could be, they could not allow a service designer to be exhaustive in the stage one completion and in the spotting of interaction. Experience teaches that all the possible goals of a service may not be foreseen by a service designer, all the more so the tricky policy interaction cases.

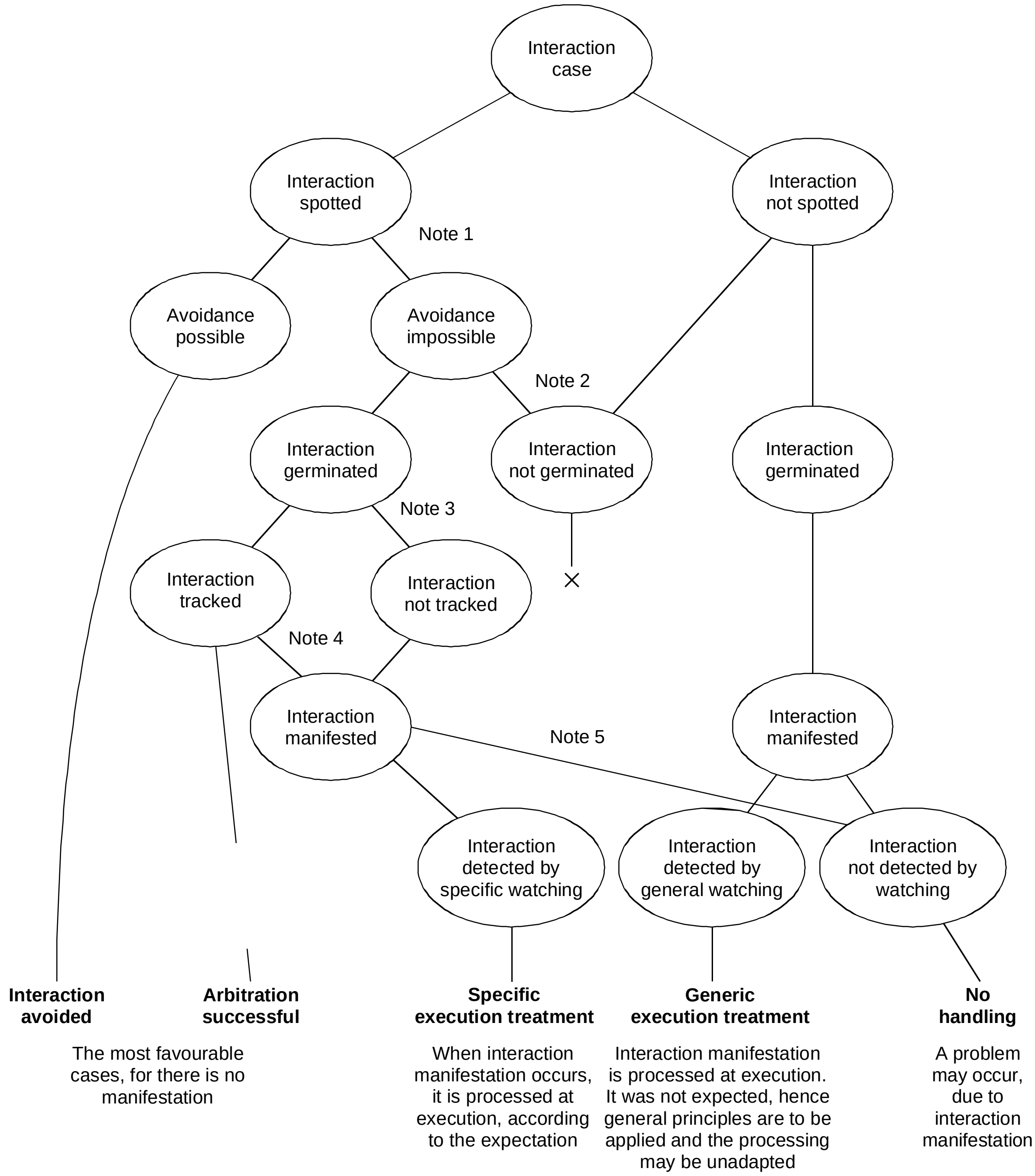
A mechanism in call processing is thus necessary, and spotting results may lead to adapt the execution arbitrator. Triggered by a watching entity, execution arbitration (similar to a curative medicine absorption) aims at resolving interaction at invocation, i.e. when its manifestation occurs. It takes care of cases for which germination can not be detected (even if it has been spotted) and for which a management arbitrator would be either less efficient or too costly with respect to the real problem. It might be extended to unspotted interaction cases.

Here as well, some feedback is to be given to the SCE, in order to provide information on the problems encountered during service registration or activation.

Such an interaction management at service execution level concerns the distributed functional plane and is developed in Recommendation Q.1204. Two main approaches, the feature interaction manager and the negotiation concept, are described.

2.3.4.3.4 Recapitulation of the possible scenarios

Figure 4 summarizes the possible ways to solve an interaction case, indicating the probability of satisfaction associated to each of the scenarios.



Probability of satisfactory interaction resolution			
CERTAIN	VERY HIGH	HIGH	MEDIUM to LOW

T1184360-97/d05

Figure 4/Q.1202 – Possible scenarios for interaction resolution

Notes relative to Figure 4:

NOTE 1 – Avoidance techniques do not apply.

NOTE 2 – For these call instances, data configuration was such that no interaction occurred.

NOTE 3 – Germination cannot be detected.

NOTE 4 – The service designer's choice is to intervene at service execution. It may occur either when management arbitration is too complex or too costly, or when processing at execution is foreseen to be sufficient.

NOTE 5 – Interaction has been spotted, but no handling has been implemented. It means that the service designer's choice was not to treat interaction, e.g. because the cost of the treatment is not worth the candle compared to the minor consequences of the manifestation.

2.3.4.4 Automating and standardizing interaction management

As seen above, it is not difficult to imagine several solutions to resolve an undesired interaction. Furthermore, it appears clearly that interaction is to be taken into account as soon as its detection is possible. Two main problems remain:

– between two services enriched with several features, there is a very large number of possible interactions, to avoid considering all possible cases, interactions shall be resolved on a case-by-case basis;

– in open network scenarios, service designers having shallow knowledge of the network and no responsibility to preserve its efficiency could be empowered to define how the network is to handle their calls.

The solution of these two problems requires:

– a high level of automation, both in the computerized spotting of the possible desired interactions themselves and in the setting of automatic mechanisms that allows the managing of the interaction occurring during processing and charging of the calls;

– a good level of standardization and a **coordination authority**, to allow the interworking of complex services over different networks, or in the same network over different service providers[2].

The coordination authority could manage an administrative environment to aid concurrent, independent service development by screening developers from implementation level conflicts with other services or service features. For example, this administrative environment should support assignment of global attributes, library functions available to developers, monitoring facility for all global system data, abstract data definitions, etc. It should also receive feedback from the interaction arbitrators and from other network sources, in order to be able to formulate recommendations regarding the base of rules and the service descriptions (see Figure 5).

The achievement of this authority may be difficult:

– Certification of a service, i.e. guarantee that all possible interactions are identified and handled, is likely to be an issue.

– It may be technically difficult to come to a common understanding on the principles to be applied to the policy interaction cases.

– In addition to the technical difficulties, the service providers, especially in case of competition, may consider that information on planned services is sensitive.

Coordination is required for each service addressing domain. International standardization bodies can take care of the technical aspects of this coordination for the international services.

---

[2] It is to be observed that interaction is not only related to service call processing, but also covers service charging. For instance, when a mobile makes a call, the radio part is charged to the mobile account, even if the called number is a Freephone number; for the caller, the call is not totally free any more. If the service provider wants this radio part to be charged to the Freephone account, a charging interaction management is necessary.

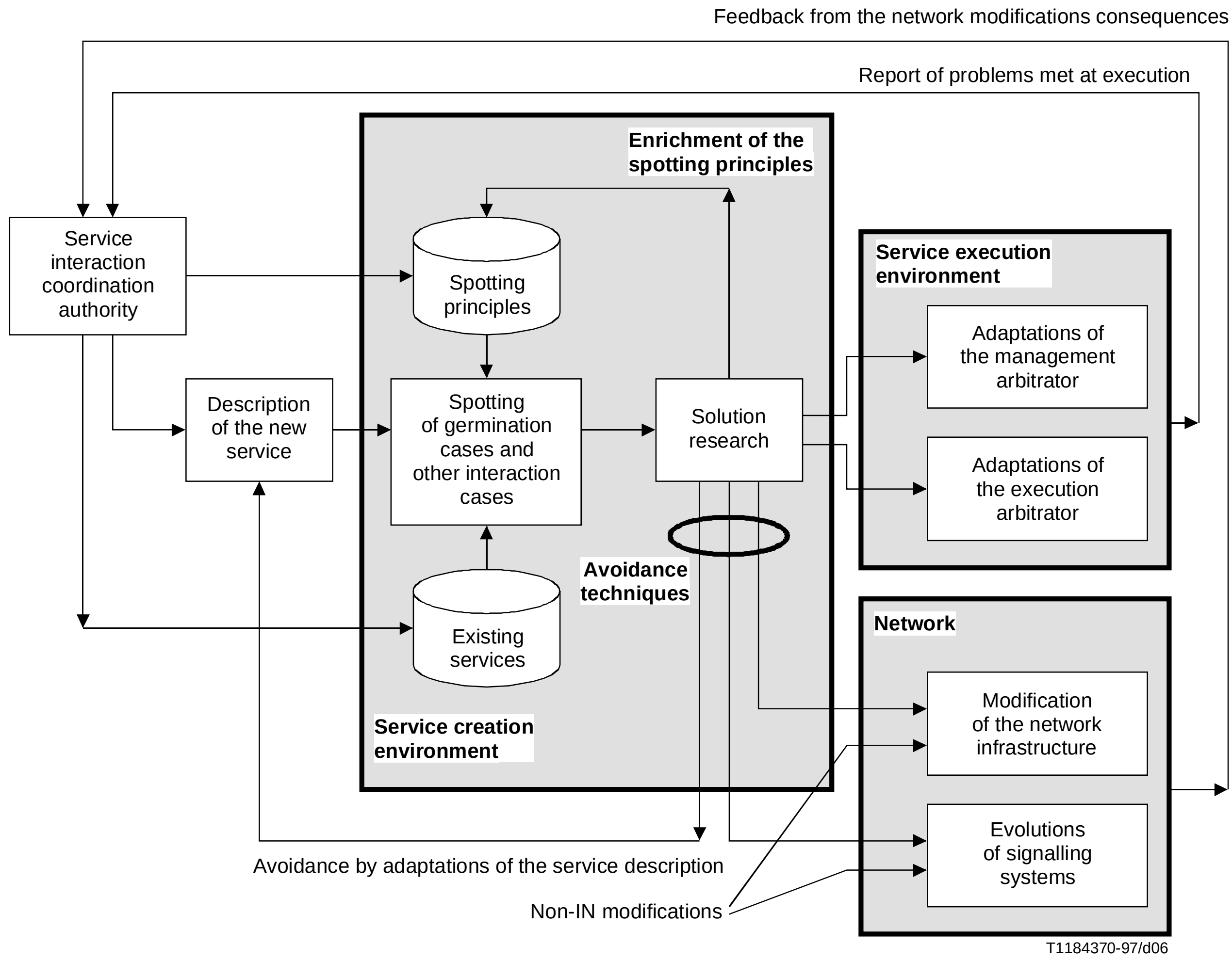


Figure 5/Q.1202 – General scheme of interaction management in a multi-provider environment

Annex A

Alphabetical list of abbreviations used in this Recommendation

CLIP	Calling Line Identification Presentation
CLIR	Calling Line Identification Restriction
CUG	Closed User Group
IN	Intelligent Network
INCM	Intelligent Network Conceptual Model
SF	Service Feature
SIB	Service Independent Building Block

Annex B

A thesaurus of examples

B.1 General

This Annex intends to provide some basic examples of service and feature interaction.

B.2 Call Forwarding Unconditional and Terminating Key Code Screening

B.2.1 First case

The Terminating Key Code Screening service enables a subscriber to have incoming calls screened by a user-defined key, i.e. key code. Callers are required to enter this key. The subscriber may activate and deactivate the service.

Consider the interaction between Call Forwarding Unconditional and Terminating Key Code Screening, when a user wants to make his calls forwarded to a line screened by a key code.

This situation is an interworking case: the caller A could be connected to an announcement asking him to compose a key code he his not aware of, because he does not know the line he gets through to (see Figure B.1).

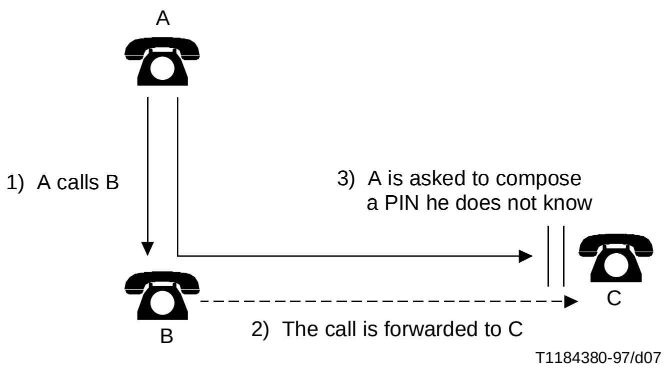


Figure B.1/Q.1202 – Invocations of Call Forwarding and Terminating Key Code Screeing on two different lines

B.2.2 Second case

Considering the same pair of services, if a user has his line screened by a key code and makes his calls forwarded, it is to be decided whether the call forwarding is to be considered prior to the key code screening or not (see Figure B.2).

B.3 Call Forwarding Unconditional and Automatic Call Back

B.3.1 First case

Automatic Call Back allows a user using a control procedure to call back the calling party of the last incoming call.

Consider the interaction between Automatic Call Back (ACB) and Call Forwarding Unconditional (CFU). If a user B has requested a CFU to C, and if a user A places a call to B, the call is forwarded to C, who may use ACB against A. Interworking is acceptable: if he uses ACB, C gets through to the user who placed to previous call. One may consider that there is no interaction problem (see Figure B.3).

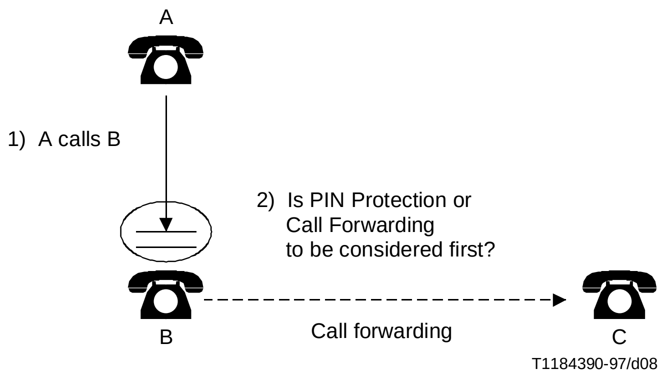


Figure B.2/Q.1202 – Invocations of Call Forwarding and Terminating Key Code Screeing on the same line

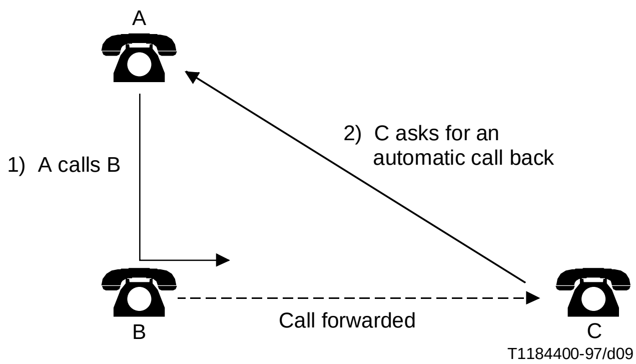


Figure B.3/Q.1202 – Invocations of Call Forwarding then Automatic Call Back

B.3.2 Second case

However, if B calls A, and if A tries to use ACB against B, the call would be forwarded to C that will probably will not be aware of the call and may not understand the situation (see Figure B.4). Interaction solution is not obvious, and several possibilities may be envisaged:

1) The call back may "normally" be routed to C's line; this plain solution may disturb C who doesn't know where the call may come from (but it is the best if B and C lines are owned by the same person).

2) The call back may be routed to B's line, where A's caller is located; this choice is the best from A's point of view, for it tries to make the return call as efficient as possible; one may also consider that B wants to get in touch with A, since B has just called A, which justifies CFU's overriding.

3) One may mix previous solutions, ringing B first (where the caller is probably located), then C if B doesn't answer; in such a case, the CFU is converted in CFNR.

4) One may forbid the calls generated by ACB if the calling party has activated a Call Forwarding.

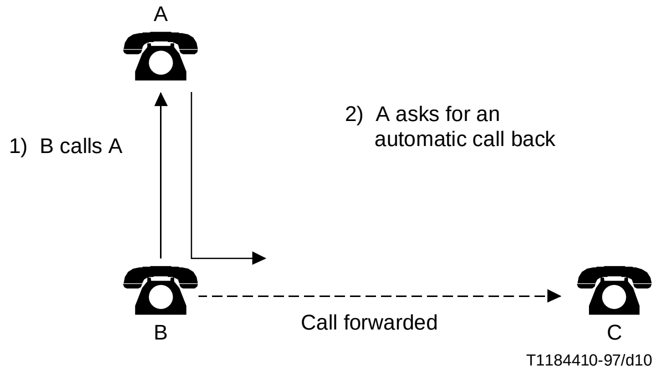


Figure B.4/Q.1202 – Invocations of Automatic Call Back then Call Forwarding

B.4 Call Forwarding Unconditional and Terminating Call Screening

The case of interworking between Call Forwarding Unconditional and Terminating Call Screening is complex, because in each service, several users may be found:

– the CFU users are "caller", "called" and "forwarded-to";

– the TCS are "screener" and "screened".

The permutation reveals that interworking between these two services presents 36 different cases to scan, among which six (Nos. 19 to 24) may be merged. Table B.1 shows the results of users permutation, considering the scenario "B forwarding his calls to C".

Table B.1/Q.1202 – Interworking between TCS and CFU

No.	"Screened"	"Screener"	Caller	Calling	Result
1			A	B	No impact, call rejected
2				C	No impact
3	A	B	B	A	No impact
4				C	No impact
5			C	A	No impact
6				B	If CFU authorizes call back from C to B, the call is completed. Otherwise, busy tone is sent to C.
7			A	B	Call is forwarded, but rejected when terminating at C
8				C	No impact, call rejected
9	A	C	B	A	No impact
10				C	No impact
11			C	A	No impact
12				B	If CFU authorizes call back from C to B, the call is completed. Otherwise, busy tone is sent to C.
13			A	B	No impact, call forwarded
14				C	No impact
15	B	A	B	A	No impact, call rejected

Table B.1/Q.1202 – Interworking between TCS and CFU *(concluded)*

No.	"Screened"	"Screener"	Caller	Calling	Result
16				C	No impact
17			C	A	No impact
18				B	If CFU authorizes call back from C to B, the call is completed. Otherwise, busy tone is sent to C.
19 to 24	B	C			If B wants to activate a CFU to C whereas C has put him in his blacklist, the services will include incompatible data.
25			A	B	(See Figure B.5) The call is forwarded, but A may be connected to C against his will because C is in A's blacklist; therefore A has all the best reasons to avoid C.
26	C	A		C	No impact
27			B	A	No impact
28				C	No impact
29			C	A	No impact, call rejected
30				B	If CFU authorizes call back from C to B, the call is completed. Otherwise, busy tone is sent to C.
31			A	B	No impact, call rejected
32				C	No impact
33	C	B	B	A	No impact
34				C	No impact
35			C	A	No impact
36				B	No impact, call rejected

Case No. 25 shows a very tricky case which would probably escape a heuristic investigation (see Figure B.5). User B having forwarded his calls to C, A is a user who has the best reasons to avoid C. If A calls B, the CFU will make the call forwarded to C, and A and C will be linked. If it is considered to be avoided, one may state that a call is not to be forwarded to a line which is in caller's blacklist.

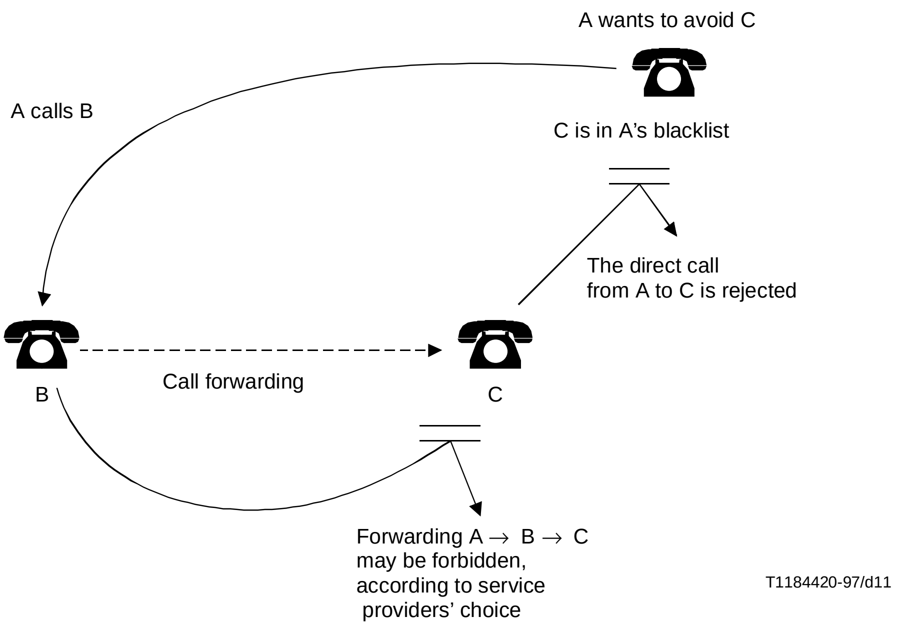


Figure B.5/Q.1202 – Complex interaction between Call Forwarding Unconditional and Terminating Call Screening

ITU-T RECOMMENDATIONS SERIES

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