ETSI TECHNICAL REPORT

ETR 323

December 1996

Source: ETSI TC-NA

Reference: DTR/NA-006007

ICS: 33.020

Key words: IN, service

Intelligent Network (IN); Service life cycle reference model for services supported by an IN

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 4 92 94 42 00 - Fax: +33 4 93 65 47 16

Contents

Blank page

Foreword

This ETSI Technical Report (ETR) has been produced by the Network Aspects (NA) Technical Committee of the European Telecommunications Standards Institute (ETSI).

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 the application of ETSs or I-ETSs, or which is immature and not yet suitable for formal adoption as an ETS or an I-ETS.

This ETR is based on CCITT Recommendations Q.1201 [1] to Q.1290 [2] as given in CCITT COM XI-R 164, 1992.

Introduction

This ETR provides a framework for the description of the Service Life Cycle (SLC). The SLC is the description of both phases and activities involved during the complete life of any service, in a service independent manner. It is considered the basis for defining the possible behaviour of a service at all times, the phases identified covering all aspects of a service life, including its "death". The SLC is composed of several phases, with specific activities that can be carried out by one or more actors in each phase. It is intended that this framework applies to all possible services. This implies that in some cases some of the activities may be not applicable.

The SLC may be viewed in two perspectives:

a) to describe the processes that the different actors have to face in order to offer a new service in a network. For instance, for a service provider there is a difference between offering generic services and subscriber specific services;

b) to describe the different states that occur during the execution of a service (such as "subscribed", "activated", "invoked"). This can be relevant for identifying service interactions, or for identifying how the service subscriber experiences the service when it is being used.

These different views are outlined in clause 4.

Blank page

1 Scope

This ETSI Technical Report (ETR) develops the concept of Service Life Cycle (SLC) for all kinds of services supported by an Intelligent Network (IN). However, this ETR could be used in a broader perspective. Notably, the used terminology is in line with that used for Integrated Services Digital Network (ISDN) supplementary services.

The SLC is the description of both phases and activities involved during the complete life of a service, in a service independent manner. During each phase activities are identified that can be undertaken by actors such as service providers, network operators, service subscribers, etc.

The rationale for defining a SLC reference model and its intended usage is:

- it creates a **common understanding** and a **common terminology** to be used when discussing a service;

- it provides a **common framework** that structures the discussion on many other issues, such as IN security requirements given in ETR 320 [5], IN management requirements given in ETR 319 [6], service interaction aspects given in ETR 137 [7] and service creation aspects;

- it helps to identify where in the service creation process additional support is needed (tools, methodology);

- it helps to identify and structure internal processes involved for a service provider or network operator when offering services.

In this ETR, the SLC is not described to its fullest extent. Instead, a framework is given that can be refined when needed.

2 References

This ETR incorporates by dated and undated reference, provisions from other publications. These references are cited at the appropriate places in the text and the publications are listed hereafter. For dated references, subsequent amendments to or revisions of any of these publications apply to this ETR only when incorporated in it by amendment or revision. For undated references the latest edition of the publication referred to applies.

[1] CCITT Recommendation Q.1201 (1992): "Principles of intelligent network architecture".

[2] ITU-T Recommendation Q.1290 (1993): "Glossary of terms used in the definition of intelligent networks".

[3] CCITT Recommendation I.130 (1988): "Method for the characterization of telecommunication services supported by an ISDN and network capabilities of an ISDN".

[4] ETR 322: "Intelligent Network (IN); Vocabulary of terms and abbreviations".

[5] ETR 320: "Intelligent Network (IN); Security requirements for global IN systems".

[6] ETR 319: "Intelligent Network (IN); IN intra domain management requirements for Capability Set 2 (CS2)".

[7] ETR 137: "Intelligent Network (IN); Service and feature interaction: service creation aspects, service management aspects and service execution aspects".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of this ETR, the following definitions apply:

actor: See ETR 322 [4].

generic services: Services created to suit the needs of a variety of service subscribers.

network operator: See ETR 322 [4].

service instance: A particular combination of service data and service logic that applies to only one service subscriber.

Service Life Cycle (SLC): The description of both phases and activities involved during the complete life of any service, in a service independent manner.

service provider: See ETR 322 [4].

service subscriber: See ETR 322 [4].

service type: A collection of functions and data distributed across network resources, providing the potential for the offering of a service instance to a customer.

service user: See ETR 322 [4].

NOTE: In the context of open networking, the terms service user, service subscriber, network operator, service provider and customer, are currently being reconsidered; additional terms (e.g. service broker, information provider) are being discussed.

subscriber specific services: Services created to initially suit the need of a single service subscriber only.

3.2 Abbreviations

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

IN	Intelligent Network
ISDN	Integrated Services Digital Network
SLC	Service Life Cycle

4 Service Life Cycle: phases and activities

The SLC consists of a series of phases. During a phase, certain activities can be performed, either in sequence or in parallel. Activities are assembled into activity groups. The identified phases are:

- needs analysis;
- service creation;
- service acceptance testing;
- service deployment;
- service provisioning and operation;
- service removal.

Any service can be described at three different levels: service type, service instance and service invocation:

- **service type related activities** involve all general aspects of the service. This covers the following phases: needs analysis, service creation, service acceptance testing, service deployment and service removal;

- **service instance related activities** involve all aspects of a service type associated with a given service subscriber. This covers activities of the service provisioning and operation phase, which can only take place when a service type is deployed in the network;

- **service invocation related activities** involve all aspects that influence the invocation(s) of a given service subscriber's service instance. These activities can only take place after service subscription and provisioning.

The phases and activities of the SLC are presented in figure 1. For the activities within the service provisioning and operation phase, any alteration of control data within a control activity will affect all activities shown to its right within the diagram. Namely, any use of service instance control will be reflected in all invocations of that instance, while any service type control actions will affect all instances (and therefore invocations) of that service.

If a problem occurs within a phase of the SLC, resulting in an inability to proceed to the next phase, it is possible for the preceding phase to be re-entered in order to take any necessary corrective measures. For example, if the service type creation fails in the service acceptance testing phase, it is possible to return to the service creation phase to re-evaluate any design flaws.

NOTE: After provisioning and subscription, an instance of the service type is created. So, for one service type, multiple instances may exist at the same time. In the same sense, many service invocations may occur within the context of one service instance.

PHASES	ACTIVITY GROUPS			
Needs analysis	**- market needs analysis** **- customer needs analysis**			
Service creation	**- service specification** **- service validation** **- service type development and building block development** **- service verification**			
Service acceptance testing	**- service acceptance test** **- service pilot (field trial), ...**			
Service deployment	**- downloading of service logic and service data into network elements** **- installation of management functions and management data associated with the service**			
Service provisioning and operation	**Service type control activities**	**service instance provisioning**		
		preparation activities	**activation activities**	**invocation activities**
		service instance withdrawal		
Service removal	**- removal of service logic and service data from network elements** **- removal of management functions and management data associated with the service**			

Service type related activities
Service instance related activities
Service invocation related activities

NOTE: Activities shown to the right occur within the context of activities shown to the left. Activities shown below another activity are only allowed if the upper activity has already taken place. The shading of the background indicates if the activities relate to a service type, a service instance or a service invocation.

Figure 1: Phases and main activities of the SLC

NOTE: Planning and co-ordination activities need to be carried out by a service provider to guard the progress of a service in the SLC. These activities relate to planning for development, testing, deployment, etc.

In each of the following subclauses, a phase is described, as well as the activities within that phase. The relevant actors performing the activities are also identified.

In each phase, activities related to service interaction can be identified (e.g. service interaction spotting during the service creation phase, or service interaction germination and service interaction watching in the service provisioning and operation phase). Such activities are defined in ETR 137 [7] and are outside the scope of this ETR.

4.1 Needs analysis phase

The initiative to create a new telecommunication service which is supported by IN capabilities may be taken either by the service provider (driven by needs detected in the market or needs detected for a particular customer) or by a customer. From the creation point of view, the following types of services are to be distinguished:

- **generic services**: services created to suit the needs of a variety of service subscribers. The creation of generic services is market driven and, when deployed, the services may be offered to all customers without modification or may be parameterized for the needs of a specific service subscriber; i.e. a service instance unique to the customer is generated;

- **subscriber specific services**: services created to initially suit the need of a single service subscriber. This may be considered as a joint initiative between the service provider and the customer, which occurs when the requirements of the customer cannot be met through the parameterization of an existing generic service, resulting in the need for the creation of a new service type. However, this subscriber-specific service has the potential to become generic if, during the creation process, parameterization options are included.

If an individual customer requires the coverage of particular telecommunication needs, then these specific needs are analysed. Initially, they should be examined to evaluate if they can be met by provisioning a generic service, whereby further customization may occur during the service instance provisioning activities of the service provisioning and operation phase, by means of selecting appropriate parameter settings (service instance parameterization). If the customer's requirements cannot be met in this way, then the creation of a subscriber-specific service may result. This process is reflected in the flowchart shown in figure 2.

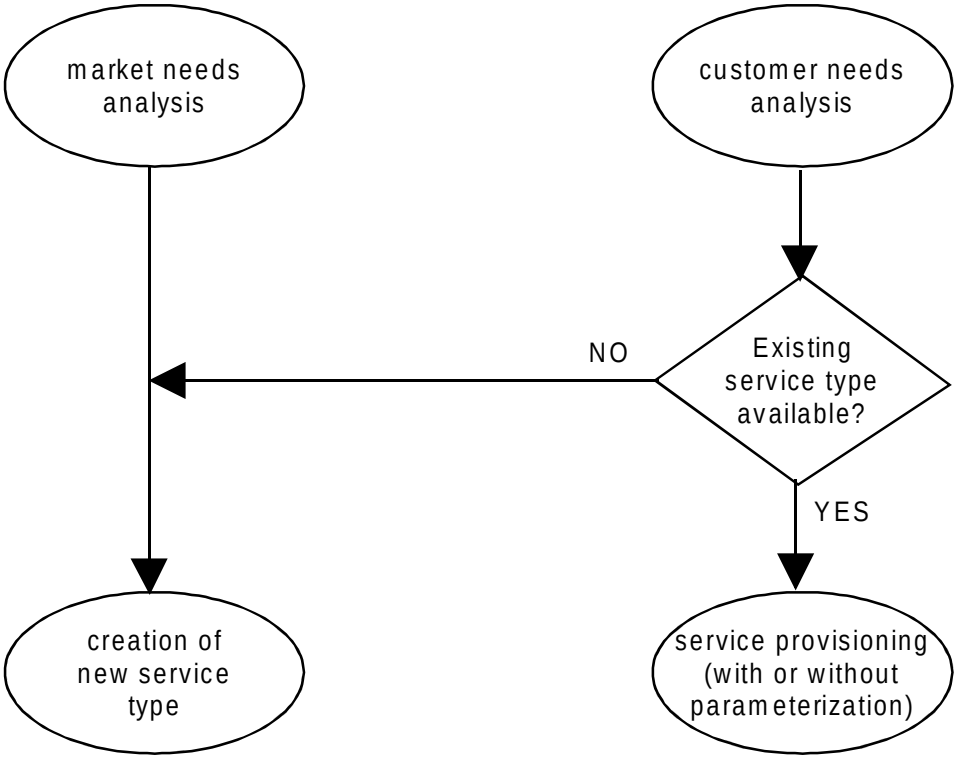

Figure 2: Flowchart diagram to illustrate when the creation of a new service type is required and when existing service types can be used

Needs analysis phase consists of two activities:

- **Customer needs analysis:** customer needs analysis is an activity where the customer and the service provider agree upon a set of requirements which adequately represents the customer's needs.

- **Market needs analysis:** market needs analysis is an activity where the service provider explores the market and specifies a set of requirements which adequately represents an foreseen need in the market.

The output of the needs analysis phase is a stage 1 description (see the three stage methodology defined in CCITT Recommendation I.130 [3]) of the service.

NOTE: **Modifying existing services:** a modified service logic is considered as constituting a new service with its own life cycle. In order to achieve a smooth transition from a service version to an updated version, it is conceivable to deploy both versions in the network, only one being enabled at a time, and, if the service is offered in a distributed way, to enable different versions in different domains (groups of service subscribers, groups of locations). Another method consists in the removal of the old version before the new version is deployed. Particular attention should be paid to the case where the service logic is to be replaced by an updated version using the same data template; in this case, it should be possible to copy the existing service data into the new service version, or simply maintain it.

Actors: service provider, service subscriber.

4.2 Service creation phase

The **service creation phase** is subdivided into the following activity groups:

- service specification;
- service validation;
- service development;
- service verification.

Actors: service provider, network operator.

4.2.1 Service specification

Service specification activities transform the service requirements into a final stage-1 service description agreed with the customer or the service provider, and the definition of a high level design by means of refinement of detailed description requirements and functional analysis.

4.2.2 Service validation

Service validation activities comprise the formal testing of the specification, in order to validate if it satisfies the needs. Testing is performed in a prototyping or simulation environment. Prototyping may provide, by means of simulation tools, Use Case Diagrams that enable a better practical understanding of the service and facilitate the preparation of the instructions given to the customer.

4.2.3 Service type development and building block development

Service type development and building block development is the transformation of the high- level design into the necessary set of software components (building blocks), data definitions, etc., required to realize that design. When the re-use of existing building blocks do not suffice to build the service type, new ones may have to be developed (as a parallel activity with the service type development).

4.2.4 Service verification

Service verification activities comprise the rigorous test of the developed software, in order to verify that the resulting service completely satisfies the specification. Testing is performed in a non-operational environment.

Service verification and validation may include specific conformance tests in order to ensure the needed openness of the service. At service specification level, specific conformance points can be specified. Such conformance points or reference points in the service system may include perceptual conformance, interworking conformance or interchange conformance.

4.3 Service acceptance test phase

The **service acceptance test phases** tests, in a real IN environment, that the service is ready for deployment (see note). It may occur in either a non-operational environment or an operational environment (field trial), or both.

NOTE: Note that in some bodies the wording acceptance test may have a different meaning.

Furthermore, the service acceptance test phase may also include a try out of a service with real service subscribers and service users, so-called *service pilot*. A service pilot also encompasses management activities such as deployment and control of the service on a limited scale in the real and operational network. All subsequent parts of the SLC as described below, including the lower levels, are then applicable. However, there may usually be restrictions with relation to the number of service subscribers and possibly also with relation to the geographical location of service subscribers and users.

Actors: initiated by service provider;
carried out by network operator, service subscriber, service user.

4.4 Service deployment phase

The **service deployment phase** comprises downloading of service logic and service data into appropriate network elements and installing the management functions and management data associated with the service. Allocation of resources is required to identify the relevant resources necessary to support the service type.

Actors: initiated by service provider;
carried out by network operator.

4.5 Service provisioning and operation phase

The **service provisioning** process covers all the activities which relate to creating service instances of a service type, preparing them for operation and eventually withdrawing them (see figure 1).

Service operation covers on the one hand all the activities which relate to the users modifying stored data (preparation activities) and invoking the service instance (possibly multiple times), and on the other hand the control of these invocations (see figure 1).

The service provisioning and operation phase is subdivided into the following activity groups:

- service type control activities;

- service instance and invocation activities.

These activity groups are evaluated in the next two subclauses. Furthermore, in a third subclause, the various states a service instance can be in during this phase of the SLC, as well as the activities that cause transitions between the states, are described in a finite state model.

The following actors may be involved in service operation:

- the **service provider** may act at service instance provisioning and subscription, at service type parameterization, at service instance parameterization, at service type control and at service instance control;

- the **service subscriber** may act at service instance parameterization and activation;

- depending on the service, the **service user** may act at registration and at invocation;

- another **service user** may act at invocation.

Actors: service provider, network operator, service subscriber, service user.

4.5.1 Service type control activities

The **service type control activities** may start after downloading of the service software into the appropriate network elements as well as the related management software into the appropriate management systems. These activities allow the modification of service parameters on a global level, i.e. those parameters which are **not** service subscriber specific. These activities are consistently performed and service data integrity is guaranteed on a network wide basis.

Only the service provider and the network operator will be enabled to perform service type control, in contrast to service instance control, which, depending on the nature of the service, may also be possible for the service subscriber.

The following activities, outlined in the subsequent subclauses, are to be distinguished during the service control:

- service type enabling / disabling;
- service type feature enabling / disabling;
- service type parameterization.

Actors: service provider, network operator.

4.5.1.1 Service type enabling / disabling

In order to allow the service type to be globally operative or not, it is necessary that it can be enabled and disabled on a network wide level.

Whether a service is enabled or disabled should not affect the possibilities of global level service control and, in the case of generic services, the possibility of setting parameters related to subscription and withdrawal. The advantage of service type enabling is allowing a service to be put into operation at a given point in time by means of a single command, whereas all control operations and parameter settings for the initial service subscribers, which require multiple commands, can be prepared before.

After service type enabling has taken place, service subscribers to whom the service has been provided can control their service (as far as the service allows) and service users can invoke the service. Service type disabling prevents the service subscribers from controlling their service and the service users from invoking it.

4.5.1.2 Service type feature enabling / disabling

Service type feature enabling/disabling is the activity performed by a service provider or a network operator to enable or disable certain service features. This may apply to features which have in advance been installed as part of the service, but which, for commercial or operational reasons, are put into operation at a later point in time.

4.5.1.3 Service type parameterization

Service type parameterization deals with setting general parameters that involve all service instances (all subscriptions). Example: the service provider or the network operator may change the allowed maximum number of retries for entering a PIN code from three to two times.

Service type parameterization should be independent of the enabled or disabled state of the concerned service type.

4.5.2 Service instance and invocation activities

This activity encompasses the set of following sub-activities:

- service instance provisioning;
- preparation activities;
- activation activities;
- invocation activities;
- service instance withdrawal.

Once the customer has subscribed to and is provisioned with the service, both preparation activities, activation activities and invocation activities may occur, until the service is withdrawn.

Actors: service provider, network operator, service subscriber, service user.

4.5.2.1 Service instance provisioning activities

Service instance provisioning is the process of creating a service instance from a service type.

The service instance provisioning activities are always initiated by a customer's need for a service. However, the path through the SLC preceding this point has either involved service creation, service acceptance testing and deployment in case of subscriber-specific services, or directly outcomes from needs analysis phase in case of generic services.

The first activities of service instance provisioning involve a technical feasibility study by the service provider in order to evaluate all issues raised by the provisioning of the service instance to the customer:

- **Service identification and configuration:** this activity is a matching process which first compares the description of the customer's needs against one or more service types to identify possible service instance solutions to those needs, and customizes then the identified service type to the customer's specific situation, by parameterizing the modifiable aspects of the service type.

- **Procure component service:** this activity involves the acquisition of services or service features from another service provider, in order to add value to this procured services or service features as components of another service offering.

- **Resource assignment:** the resource assignment activity identifies the resources necessary to support the requested service instance. This activity needs to consider both customer's and service provider's requirements, to ensure that both are satisfactorily met.

The next actions in service provisioning involve the actual signing of the contract between the service provider and the customer, and, additionally, the setting of certain customer-specific service instance parameters. Also, commissioning and test activities are carried out before the service instance is enabled.

- **Service subscription:** a formal agreement is drawn up, and a contract is signed, between the customer and the service provider. The customer becomes service subscriber.

- **Service instance parameterization:** service instance parameterization involves the setting of certain service parameters in such a way that the created service instance fulfils the service subscriber's particular needs. By means of service customization, the service may be further tailored to the subscriber's needs. Service instance parameterization may not be applicable to all services (e.g. call forwarding unconditional do not require such an activity).

- **Service instance commissioning and test:** specific deployment activities are necessary to ensure that the appropriate logical and physical resources are in place and operational for the service activation and subsequent use by the service subscriber. If these specific resources are not in place, they need to be deployed.

- **Service instantiation:** this activity enables the service subscriber to use the service instance as defined in the contract.

Actors: initiated by service subscriber,
carried out by service provider, network operator.

4.5.2.2 Preparation activities

By means of preparation activities occurring after service instance provisioning, the state of a service instance, with respect to a certain service subscriber, can be changed by the modification of stored service data. These activities can either be carried out by the service provider (or the network operator) or optionally, for some services, by the service subscriber. Preparation activities affect the parameter settings of the concerned service instance. The following activities are part of this activity group.

- **Registration:** registration is the programming of information enabling subsequent operation of the service instance. This programming action involves input of user-specific information. Examples: entering the forwarded-to number in case of a call forwarding service, or changing data in the user's

profile. Registration may also involve the modification of already registered data. Registration can be a part of the customization of a service.

- **Erasure:** erasure is the deletion of information stored (by a previous registration activity) for a given service instance.

  NOTE 1: With the three aforementioned activities, the service subscriber is able to modify stored data related to the service (e.g. related to his service profile). The actual modification of the data can be done in two ways: either by a management process (controlled by the service provider or the network operator, but initiated by the service subscriber), or through the execution of a part of the service logic dedicated to this purpose.

- **Interrogation:** an interrogation is the procedure of requesting information about the stored service data in the network.

  NOTE 2: In subclause 4.5.3, the relations between (sub)states and activities that cause transitions between states are evaluated in a finite state diagram.

Actors: initiated by service subscriber;
carried out by service provider, network operator (or automatically).

4.5.2.3 Activation activities

Activation activities simply enable or disable invocation of the service.

- **Activation:** service activation is the action where the service instance is activated, i.e. made ready to be invoked.

- **Deactivation:** service deactivation is the action cancelling the activation.

- **(Registration):** this action is similar to the registration action defined as a preparation activity, except that it occurs when the stored service instance data are modified without deactivating the service instance. From a modelling point of view, the service is deactivated, data are modified then the service is activated again.

- **interrogation:** an interrogation is the procedure of requesting information about stored service data in the network.

Actors: initiated by service user,
carried out by service provider, network operator (or automatically).

4.5.2.4 Invocation activities

Invocation activities starts with a service invocation request and usually ends when a certain point in the basic call handling process is reached or upon a user's action. The precise contents of the service invocation depend on the nature of the service concerned and are described as the normal and exceptional procedures for service operation in the stage 1 description of the service.

- **Invocation:** invocation is the detection of a telecommunication event starting the execution of the service instance. Such an event may either be an action taken by the user or an action automatically generated by the network or the terminal as a result of a particular condition.

  NOTE: When the service instance is invoked, the service logic may either be executing (and having control over the call processing) or waiting for a particular event to occur.

- **Invocation disabling:** invocation disabling is the action taken by the user on a per call basis to prevent the action of a service (e.g. Calling Line Identification Restriction (CLIR), suppression of Call Waiting indication during a modem call).

- **End of service execution:** end of service execution is the return of the service to the activated state, waiting for a new telecommunication event.

  NOTE: Each service invocation yields output to billing and service monitoring processes. These processes, however, are considered not to be part of the service itself.

Actors: initiated by service user.

4.5.2.5 Service instance withdrawal

Service instance withdrawal: service withdrawal is the cancellation of the contract entered by the service subscriber with the service provider. The service instance is made unavailable for the service subscriber and is deleted.

Actors: initiated by service subscriber (or service provider),
carried out by service provider, network operator.

4.5.3 State model for the service provisioning and operation phase

During this phase of the SLC, the service can be in certain stable states; the activities applicable may cause transitions from one stable state to another. In this respect, the service instance may be considered as a finite state machine[2)], as described in the next subclauses.

4.5.3.1 States

During its operation, a service instance goes through stable **states** as follows:

- **Unprovided:** the service instance has not yet been provided to the service subscriber. Note that the service type has already been deployed and is ready for provisioning and operation.

- **Provided:** the service instance has been provided to the service subscriber (i.e. a service instance has been created and is available to the user).

- **Registered:** user specific data related to the service instance have been entered.

- **Active:** the user has requested the service instance to be ready for its automatic invocation.

- **Invoked:** the service instance is being executed (either after the user has directly placed a request to the service instance or after a telecommunication event has been detected).

  NOTE: A state can not be skipped; all states have to be passed. However, in some cases such passing of states is not relevant to the service and might be considered as being performed implicitly (e.g. call forwarding: registration and activation can be performed in one action, when the user enters the forwarded-to number).

4.5.3.2 Activities that cause state transitions

The activities mentioned in the previous subclauses cause transitions between the stable states. The activities are (see subclause 4.5.2):

- service instance provision;

- service instance withdrawal;

- registration;

- erasure;

---

2) Such finite state machine may be used during the service creation activities, for interaction spotting purposes (see ETR 137 [7]).

- activation;
- deactivation;
- interrogation;
- invocation;
- invocation disabling;
- end of service execution.

Figure 3 shows which activities cause transitions between which states.

4.5.3.3 Finite state machine

Figure 3 shows a finite state machine based on the definitions of the previous two subclauses.

Each invocation of a service instance, when going from one state to another via a transition, takes its course in the finite state machine. This course may change from a service invocation to another (see above) and also from a service instance to another (e.g. a service may have been activated then deactivated without having been invoked, or invoked several times during the same activation). Therefore, we have to distinguish between the possible transitions allowed for a service instance and the actual transitions during the life of a service instance.

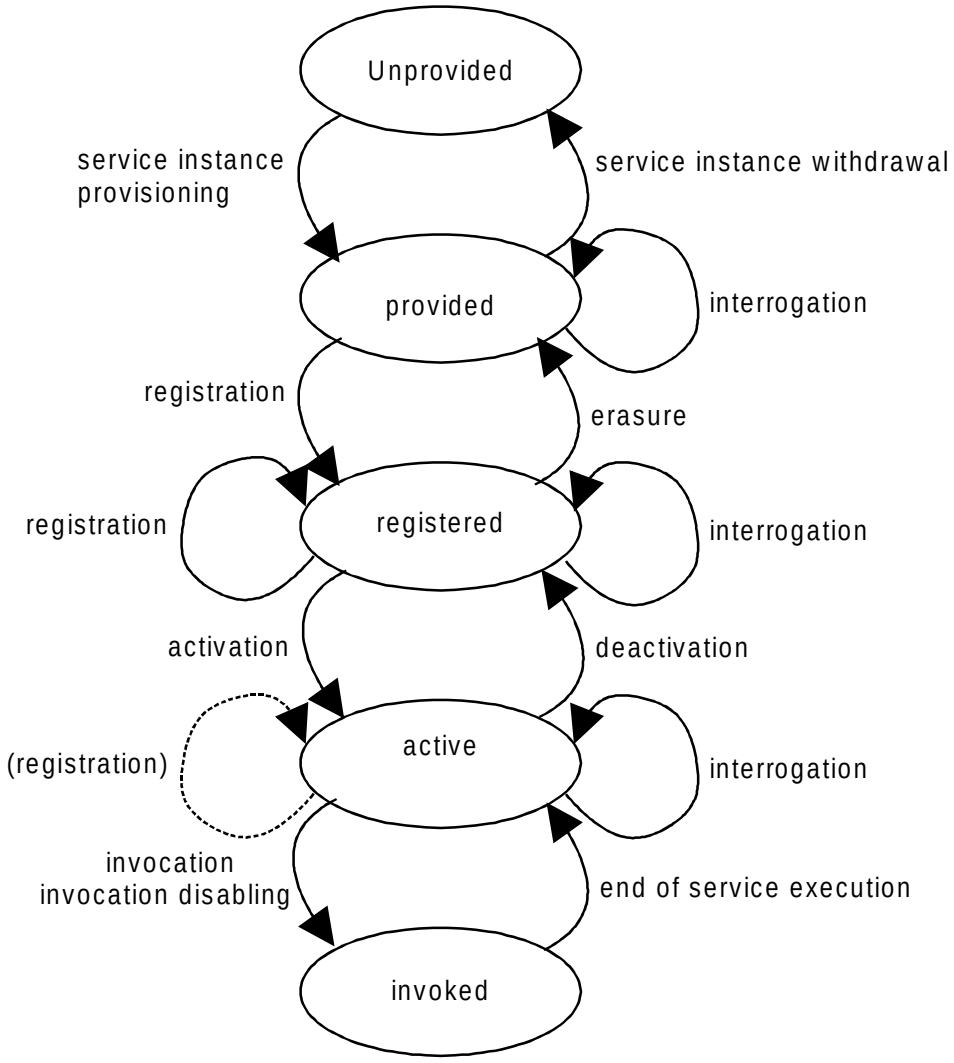

Figure 3: Finite state machine of a service instance

4.5.3.4 Examples

In this subclause, some examples on the finite state machine are presented.

EXAMPLE 1: **Card calling.**

Provided: the service instance has been made ready for use by a service subscriber, i.e. data necessary to the service instance (card number, etc.) have been recorded by the service provider for the service subscriber.

Registered: for a simple Card Calling service, no subscriber's registration activity is necessary to enter this state. However, some services require the subscriber to change his PIN: this is registration.

Active: the service user is enabled to invoke the service instance (possibly after service subscriber's action).

Invoked: a call has been placed to the service by the service user, who is requested to authenticate and to enter the called number.

EXAMPLE 2: **Call forwarding.**

Provided: the service instance has been made ready for use by a service subscriber.

Registered: the service subscriber has entered data related to the forwarded-to number. In the plainest design of the service, the registration is done in the same move as the activation. With a most user-friendly service definition which permits the recording of directory numbers associated with an abbreviated number (e.g. a digit), the registration consists in entering the directory number and the associated abbreviated number.

Active: the service subscriber has requested his incoming calls to be forwarded to a given number, and the request has been accepted.

Invoked: a call has been placed to the service subscriber and is forwarded.

EXAMPLE 3: **Terminating call screening.**

Provided: the service instance has been made ready for use by a service subscriber, but no directory number is recorded in his white list or blacklist yet.

Registered: directory numbers are recorded in the service subscriber's white list or blacklist.

Active: the white list or the blacklist has been put in operation by the service subscriber.

Invoked: a call has been placed to the service subscriber, and is either rejected or accepted according to Calling Line Identification.

EXAMPLE 4: **Completion of calls to busy subscriber.**

Provided: the service instance has been made ready for use by a service subscriber.

Registered: the data related to the outgoing call placed by the service user are recorded by the network.

Active: the service user has encountered a busy condition (on one or several calls) and asked to be informed when (one of) the called party becomes free.

Invoked: the service execution (i.e. informing the service user that the called party has become free, and setting up the new call) has started.

4.6 Service removal phase

The **service removal phase** comprises the removal of service logic and service data from the relevant network elements systems. Also the management functions and/or management data associated with the service are removed from the management systems. The service should first be disabled before the removal takes place.

These activities are consistently performed and service data integrity is guaranteed on a network wide basis.

Actors: initiated by service provider,
carried out by network operator.

History

Document history	
December 1996	First Edition

ISBN 2-7437-1148-5
Dépôt légal : Decembre 1996