INTERNATIONAL TELECOMMUNICATION UNION

# ITU-T Q.1301

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(10/95)

## INTELLIGENT NETWORK

# TELECOMMUNICATION APPLICATIONS FOR SWITCHES AND COMPUTERS (TASC) – TASC ARCHITECTURE

## ITU-T Recommendation Q.1301

(Previously "CCITT Recommendation")

## FOREWORD

The ITU-T (Telecommunication Standardization Sector) is a permanent organ of the International Telecommunication Union (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 (Helsinki, March 1-12, 1993).

ITU-T Recommendation Q.1301 was prepared by ITU-T Study Group 11 (1993-1996) and was approved under the WTSC Resolution No. 1 procedure on the 17th of October 1995.

---

NOTE

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

# CONTENTS

## SUMMARY

This Recommendation identifies the Architecture that supports TASC (Telecommunication Applications for Switches and Computers) and is one in the Q.1300-Series Recommendations on TASC. The main purpose of TASC is to allow applications running within the network user's environment to integrate telecommunication services with computing facilities. This would typically allow business applications to use TASC to integrate the computer workstation and telephone at the user's desktop. This Recommendation does not define how the elements of the architecture are implemented. Part of the definition of the TASC architecture includes the TASC objects which are the basis for the TASC Functional Services.


## INTRODUCTION

The concept of TASC, which this architecture supports, is described in Recommendation Q.1300, TASC Overview. Although the TASC architecture identifies, describes and models interactions between the TASC objects, it does not specify how they should be implemented. In addition to call and telecommunication device objects, the role of Agents (as in ACD system) is modelled.

## BACKGROUND

This Recommendation is based on the experience of ECMA (Standardising Information and Communication Systems) and ANSI (American National Standards Institute) member companies in developing switch-to-computer interfaces and takes directions from CSTA (Computer Supported Telecommunications Application) and SCAI (Switch-to-Computer Application Interface) standards.


## KEYWORDS

Architecture, Models, Objects, TASC.

**Recommendation Q.1301**

# TELECOMMUNICATION APPLICATIONS FOR SWITCHES AND COMPUTERS (TASC) – TASC ARCHITECTURE

*(Geneva, 1995)*

## 1 Introduction

This Recommendation defines the architecture for TASC and identifies the various objects which represent information which the TASC interface operates on. The concept behind TASC is described in Recommendation Q.1300, which is considered essential reading prior to this Recommendation, and the services that the architecture supports are defined in Recommendation Q.1302. Recommendation Q.1303 defines the requirements for managing the objects herein described.

## 2 Scope

This Recommendation defines an architecture for the support of Telecommunication Applications of Switches and Computers. The architecture supports the communication between the switch and computer and the services conveyed by that communication. This Recommendation considers how the information communicated is represented. TASC does not define how the aspects making up TASC in either the switch or computer environment are implemented.

The emphasis of TASC has been to define third-party call control functions which also encompasses first-party call control. TASC is independent of any underlying mechanism and is applicable to public, private and hybrid networks. TASC is designed to be flexible in order to support other communication environments in addition to those based upon ISDN and Intelligent Network (IN) principles.

It is focused on providing an application service interface between a switch and computer.

TASC supports both a single-ended call (originating and terminating) view as well as a global call view.

The TASC architecture accommodates different ways of viewing calls available in implementations but models these views on stable and accepted models of how calls work.

## 3 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision: all users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published.

- ITU-T Recommendation Q.1300 (1995), *Telecommunication Applications for Switches and Computers (TASC) – General overview*.

- ITU-T Recommendation Q.1302 (1995), *Telecommunication Applications for Switches and Computers (TASC) – TASC Functional services*.

- ITU-T Recommendation Q.1303 (1995), *Telecommunication Applications for Switches and Computers (TASC) – TASC management: Architecture, methodology and requirements.*

# 4 Terms and Definitions

This Recommendation uses terms defined in Recommendation Q.1300.

## 4.1 Abbreviations

For the purposes of this Recommendation, the following abbreviations are used:

ACD Automatic Call Distributor

CE Communication Entity

CP Communication Party

CV Call View

FS Functional Service

OCV Originating Call View

TASC Telecommunication Applications for Switches and Computers

TCV Terminating Call View

# 5 Architecture

The TASC Architecture provides a framework for communication between a computer and a switch. Recommendation Q.1300 describes the environments in which TASC may be used.

## 5.1 Domains

Domains identify the area that TASC may manipulate and influence.

### 5.1.1 Domain

Call domains ultimately deal with controllable and visible telecommunication devices in TASC's problem space. Such devices are referred to as Communication Entities (CE) so as to avoid confusion with connotations of specific types of devices.

#### 5.1.1.1 Control and visibility

TASC deals with the control and visibility of calls as they originate and terminate at CEs. All TASC functions revolve around this fundamental principle. Therefore, the problem space involves calls which can originate, terminate, or become visible at CEs. The CEs defined by TASC include Line CEs and Distribution CEs.

When dealing with a call the switch is transparent to the application. TASC does not attempt to directly control or provide visibility of switches and computers. So switch and computer objects are not in the TASC problem space. Only the CEs related to or attached to switches and computers involved in the calls are visible.

#### 5.1.1.2 Objects of interest

TASC consists of a set of functional services that provide peer-to-peer applications with a standard means of communication for the control and visibility of calls. Ultimately it is the peer-to-peer applications which know the objects of interest. Peer-to-peer applications not only have a communication application context, but they also have a common, and agreed, set of CEs in the object space. In other words, the two applications know the CEs over which they have control and visibility. This object space is either statically or dynamically defined. The locations of the objects are transparent to TASC. This is depicted in Figure 1 below.

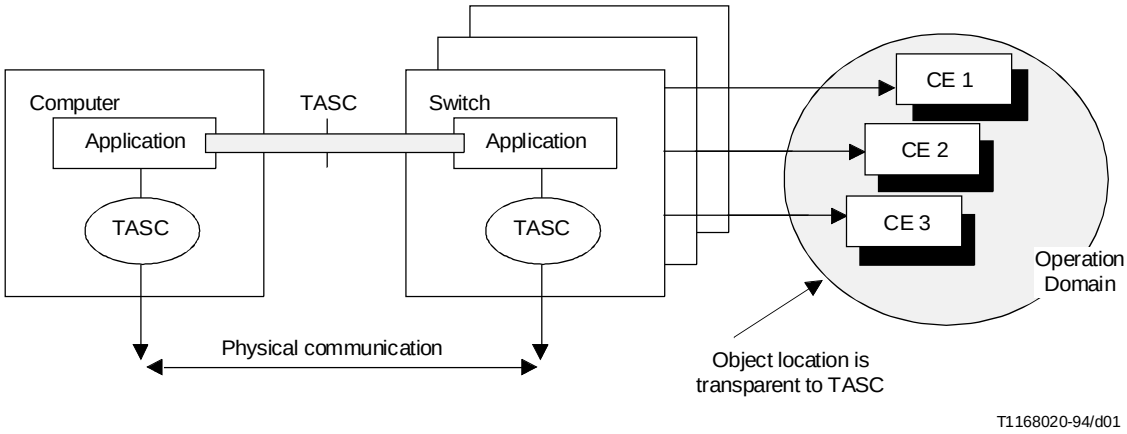


FIGURE 1/Q.1301

**Transparent Location of CEs**

#### 5.1.1.3 Operation and working domains

The identification of CEs of interest to an application may occur through static (subscription arrangements) and dynamic means (messaging). The identified CEs are the operation domain for a particular application. Once that application has begun processing, the transitional objects such as CVs (calls views), CPs (connections between CEs and calls), and Users may be created and destroyed as a result of TASC related activity. The working domain may expand or contract during the lifetime of the application but may not expand to include CEs outside of the defined operation domain. It is the set of CEs (objects) which can be monitored by an application which form a Working Domain. The relationship between Operation and Working Domain is illustrated in Figure 2.

TASC does not provide the ability for applications to acquire and lock the use of a CE for the exclusive use of a particular application process. Therefore coordination of applications and their use of CEs is outside the scope of TASC.

For example, in a customer service department for a company which has agents and an ACD system, the object space for the application is the set of objects representing the agent CEs and ACD pilot numbers where calls are originated, terminated, or become visible. This Operation Domain does not include end users (i.e. the customers' phones), external phones where service calls are originated, or the switches or computers. The objects which can be monitored are the agent CEs and pilot numbers, and these represent the domain of objects of interest in the calls. This object space is defined for, and known by, the peer-to-peer applications. The locations of the objects are transparent to TASC. The objects actually monitored form the Working Domain.

### 5.1.2 Call views

TASC defines a call view object to represent a call in the problem space. A call view has a call view identifier, a call view state, and participants or parties. The call view identifier is unique within the Operation domain. That is, in the communication between the peer applications, the Call View ID identifies a unique call. No two calls visible to the peer applications may have the same Call view ID.

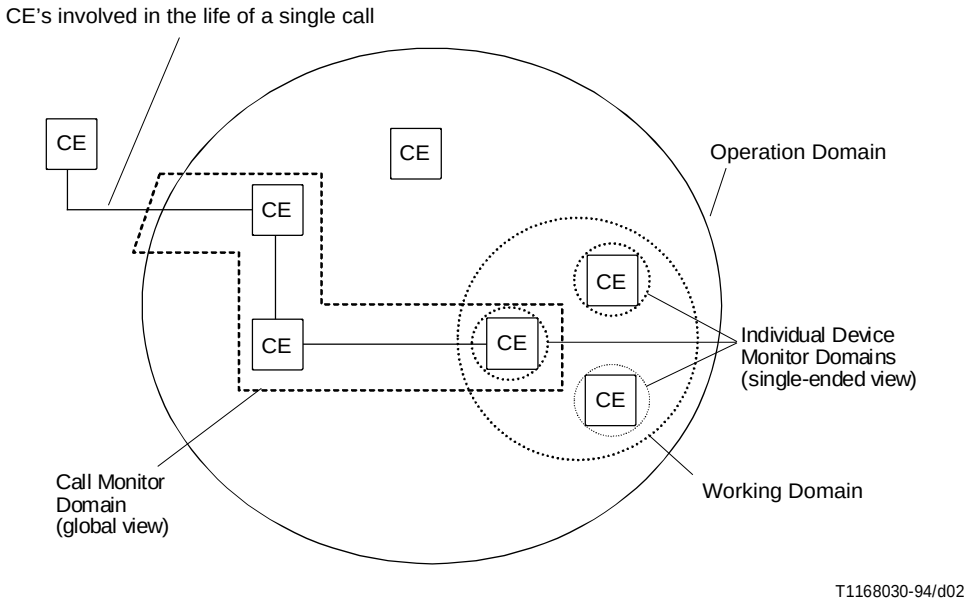


FIGURE 2/Q.1301

**Operation and working domain**

#### 5.1.2.1 Single-ended view

The single-ended view of a call is the view of call progress and call states from a CEs perspective. The TASC object model for a two-party call spits into originating and terminating views. For example, for a basic call, when that CE is originating a call, the switch provides the single-ended view incorporating the originating call model events. The actions of the terminating side are implied in those events. When that CE is terminating a call, the switch provides the single-ended view incorporating the terminating call model events. The actions of the originating side are implied in those events but cannot be guaranteed. Information available about that CE and other CEs in the call is made available in the single-ended view by the switch.

Single-ended views are independent of each other. A switch may allocate separate call view identifiers for each single-ended view.

#### 5.1.2.2 Monitoring within a single-ended view

TASC defines a single-ended view for the associated CE. This means that TASC defines call events which are reported for calls based on whether the call is originating, terminating or distributing at the CE. Thus if a CE is being monitored and a call terminates at the CE, the terminating call events may be reported. If a CE is being monitored and a call originates at the CE, the originating call events may be reported. Events may also be reported for monitored CEs at which calls appear such as Distribution CEs.

#### 5.1.2.3 Global view

An alternative to the single-ended view is the global view where the view of call progress and call states are from the domain's perspective of a call. Thus all parties associated with a specific call can be visible with a global call view. Each party associated with the call still needs to adhere to the relevant basic call model (originating/terminating or distribution) but are united by a common identifier for the call.

The extent of a global call is limited by the operation domain, thus a call that goes outside of the Operation Domain may only be able to support a single-ended view in some circumstances.

#### 5.1.2.4 Monitoring within a global view

The method used to invoke device and call monitoring differs but the event reports generated remain the same. With device monitoring, events are generated as a result of a change in the Call View state currently at the monitored device. With call monitoring, events are generated as a result of Call View state changes anywhere within the call. Two forms of Call Monitoring are defined within TASC:

1) A Call Monitor set on a call will generate events as a result of Call View state changes anywhere within the call. A call where some CEs are outside the TASC operation domain may report less call progress information; at least one CE must reside within the Operation Domain for call monitoring to function. This type of monitor will follow a single call. It may only be invoked once the call has been created and a Call View ID returned to the application, e.g. after a TASC Make Call Functional Service request or Call-Arrived event (see Recommendation Q.1302).

2) A Call Monitor set on a CE will generate events as a result of a Call View state change for every call that has involved the specified CE since the call monitor was invoked. A call where some CEs are outside the TASC operation domain may report less call progress information; at least one CE must reside within the Operation Domain for call monitoring to function. This type of monitor will follow all calls which have contact with the specified CE, e.g. a call monitor on a distribution CE will monitor all calls which are distributed by that CE. Events will be generated during the complete life of each call distributed.

Call monitoring is considered further in clause 6.

#### 5.1.2.5 Calls between CEs

As calls originate, arrive, terminate and disappear from CEs within the working domain, events are reported based on the single-ended view, features in effect, and characteristics of a CE. Since all calls are uniquely identified by a Call View ID, any and all events which specify the same Call View ID in the communication protocol between the peer applications refer to the same call. The switches assign and manage Call IDs although a management function may manipulate these ID's as described in Recommendation Q.1303

#### 5.1.2.6 Call view ID

Call lDs are assigned and managed by the switch and must uniquely identify calls at CEs. These Call lDs must be unique within the Working Domain between peer applications. How the switch assigns Call lDs to calls between CEs in the operation domain (i.e. whether or not it uses the same Call view ID for both originating and terminating objects and events) must be understood and known by the applications prior to association. Thus Call view ID assignment is part of the application rules and in part a function of the type of network – public, private or hybrid. It should be noted that a request for a global call view may be rejected based upon the capabilities of the signalling environment in which the applications exists.

#### 5.1.2.7 Global call view ID

In a call between two CEs in a working domain, a Global Call is characterized by the same Call View ID being assigned for both originating, terminating and distributing endpoints. The Call view ID is unique in the Working Domain.

Management of identifiers is achieved via parameters included in Functional Service responses and Event Reports. Identifiers cease to be valid when their context vanishes. If a call ends, its Call View Identifier is no longer valid to refer to that call.

If a call changes its Call View Identifier when a Conference or Transfer occurs, identifiers are provided to link the old Call View Identifier to the new Call View Identifier. Similarly, if a CE Identifier is changed identifiers are provided to link new and old identifiers. Event Reports are used to report state changes.

Identifiers can be re-used. Once an identifier has lost its context, it may be re-used to identify another object.

Note that it is recommended that implementations do not re-use identifiers prematurely.

Individual Call View and CE Identifiers are not guaranteed to be globally unique. TASC requires that the combination of Call View and CE Identifier be globally unique within an operation domain. To accomplish this, either the Call View Identifier, or the CE Identifier (or both) shall be globally unique.

# 6 Call monitoring

## 6.1 Introduction

Call Monitoring is a Functional Service which will provide call progress information for all Communication Entities (CEs) involved in a call. During the life of a call, regardless of the operations performed on that call, this service will continue to provide call progress information for as long as the call remains within the TASC Operation Domain. Call Monitoring will continue to provide call progress information after transfers, forwarding and conference operations.

## 6.2 CE monitoring versus call monitoring

CE monitoring will report call progress information while a call resides at CEs which have active CE monitors; this is known as the TASC Working Domain. The collection of CEs on which these monitors have been placed remains fairly static; they are not changed by any particular call. An application using CE monitoring must therefore define precisely the CEs from which it wishes to receive call progress information. The complete collection of CEs on which an application may set monitors is known as the TASC Operation Domain.

During the life of a single call many CEs may participate; some with active CE monitors and some without. An application may therefore wish to invoke a monitor which will report call progress information throughout the life of a call, regardless of the CEs which are involved in that call. Call Monitoring will provide an application with this information without having to explicitly place monitors on all the CEs which have participated in that call. With Call Monitoring, the domain of the monitor changes very dynamically to include the collection of CEs involved during the life of a particular call. Figure 2 illustrates the relationship between the various TASC domains.

The TASC Operation Domain is the collection of CEs on which an application may place a monitor. The TASC Working Domain represents the collection of CEs within the Operation Domain on which an application has placed CE monitors independently of any particular call. The Call Monitor Domain is the collection of CEs within the Operation Domain which participate in a single call when call monitoring has been enabled. There will be a Call Monitor Domain, dynamically changing as the call progresses, for each call monitor invoked. When a call dies, then its associated call monitor domain also dies. The Call Monitoring domain is only constrained by the Operation Domain and operates independently from the Working Domain but may include CEs within the Working Domain. When a CE monitor is invoked then the domain of that monitor is restricted to the individual CE, i.e. the individual CE monitor domain.

## 6.3 Call monitoring operation

TASC Functional Services such as Transfer, Conference and Forwarding (see Recommendation Q.1302) will not prevent a Call monitor from following the call even if the Call View ID changes. A Call Monitor identifier will be included in each event reported.

For a Call Monitor set on a CE a Call Monitor identifier will be returned. This Call Monitor identifier will be reported for all calls which involve that CE while the Call monitor is active. This Call Monitor identifier is distinct from any CE Monitor identifier. As an example, a Call Monitor placed upon a Distribution CE will generate call progress events for all calls which are distributed by that distribution CE and will use the same Call Monitor identifier.

# 7 TASC interfaces

## 7.1 Single TASC interface

A single viewing point covers scenarios where there is a single TASC interface. There is a single switch, a single computer, and a single application context instance. The computer thus has control and visibility of calls from a single point. However, the operation domain of CEs which can be monitored may all be local, all remote, or mixed between local and remote.

## 7.2 Multiple TASC interfaces

Multiple viewing points exist when there are multiple TASC interfaces for the same operation domain. This may involve multiple interfaces between a single switch and a single computer or multiple interfaces with multiple switches and computers. The scenario calls for a common domain of CEs known by all applications on which monitors can be placed. There are many peer-to-peer applications operating in a cooperative manner. However, in order to cooperate they all share a common definition of the operation domain which can be monitored. The operation domain of CEs which can be monitored may all be local to a single switch or distributed between switches.

There may be proprietary communication links between computers, between switches, and/or between computers and switches, to facilitate coordination and information flow. These links and information exchange are outside the scope of TASC. See Figure 3.

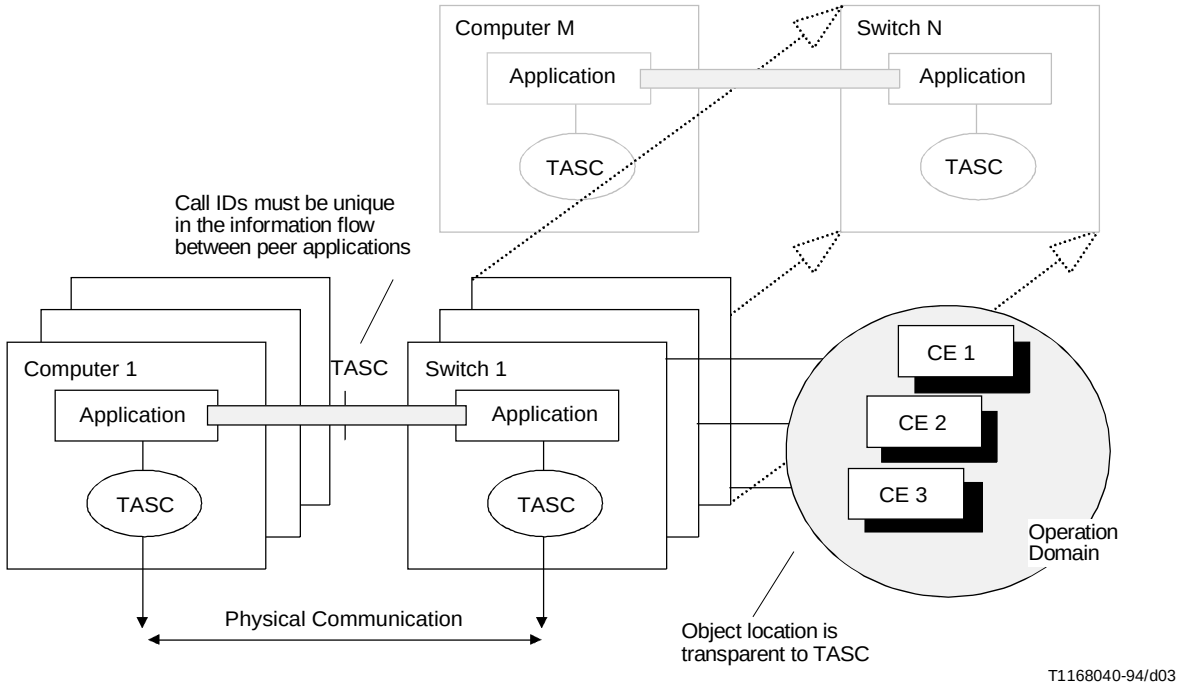


FIGURE 3/Q.1301

**Cooperating Applications**

In both ways of viewing calls identified in this Recommendation, the following is true:

A call between two CEs in the operation domain is a Global Call view ID if the same Call view ID is assigned for both the originating and terminating events.

The locations of the CEs are transparent to TASC. If a CE does not reside on the local switch, capabilities outside of TASC must exist to place a monitor on this 'remote' CE, signalling must exist to facilitate information flow between switches or computers, and global Call view ID management (i.e. management of Call view IDs between multiple switches) must be defined and provided. This is a function of the scope of the application, the type of network (public, private), and network capabilities.

In a single interface, all CEs are viewed and controlled over a single interface regardless of location. With multiple interfaces, CEs may be viewed and controlled from multiple points but the views at all points are considered with Global Call IDs.

**In summary**

1) TASC supports the following concepts: Operation Domain, Working Domains, Single-ended Calls View and Global Call Views.

2) Operation Domains are defined prior to application initiation. Working Domains are identified by dynamic means and may include a subset of those CEs which were identified as members of the Operation domain.

3) The physical locations of monitored CEs are transparent to TASC.

4) A Global Call View is a function of Call view ID management and assignment and thus a function of the switch, application, and/or network. Call view ID management is implementation specific. TASC provides the flexibility to support different Call view ID management schemes.

5) TASC does not define standardized signalling or information flow between applications in a distributed environment to facilitate coordination and control of calls across multiple TASC interfaces. However, TASC provides the basic information necessary to facilitate that coordination and control.

# 8 TASC switching model

The TASC switching model provides an abstract view of switching objects and their behaviour. This switching model consists of TASC objects, their models, and their relationships.

## 8.1 TASC objects

The following objects for TASC have been identified (see Figure 4):

**Communication Entity (CE)** – An entity that originates, terminates, or becomes visible in a call.

**User** – An entity that makes use of a CE (e.g. initiates a call or answers a call).

**Call View (CV)** – An abstraction of a call which represents the progression of a Call at a CE from the standpoint of a CE involved in a call.

**Communication Party (CP)** – An associative object that maintains the relationship between a call and CE.

### 8.1.1 Communication Entity

#### 8.1.1.1 Description and behaviour

An entity that originates, terminates, or becomes visible in a call. Many users, calls, and CPs may be associated with each CE.

#### 8.1.1.2 Types

**Line CE** – A component of a switch over which calls are originated from or terminated to a CE.

**Distribution CE** – An entity that distributes calls to other CEs.

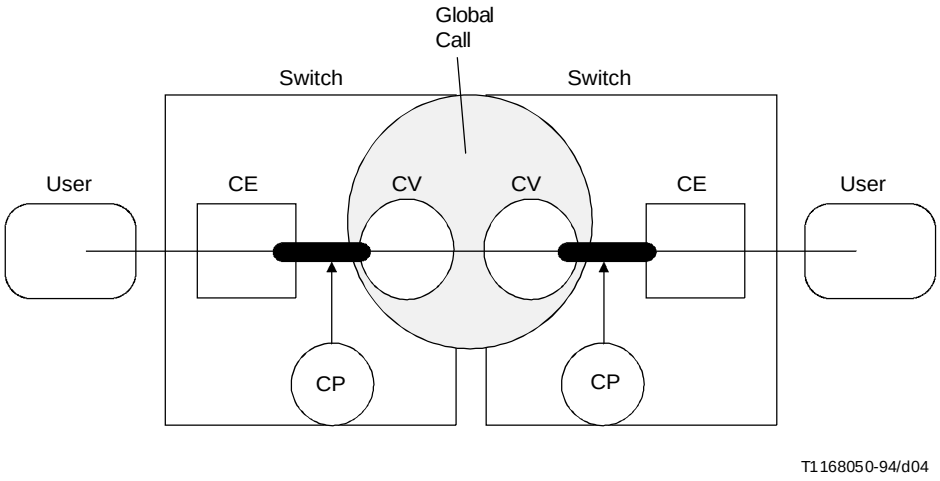


FIGURE 4/Q.1301

**TASC Objects**

### 8.1.2 Line CE

#### 8.1.2.1 Description and behaviour

A component of a switch over which calls are originated from or terminated to a user. See Figure 5.

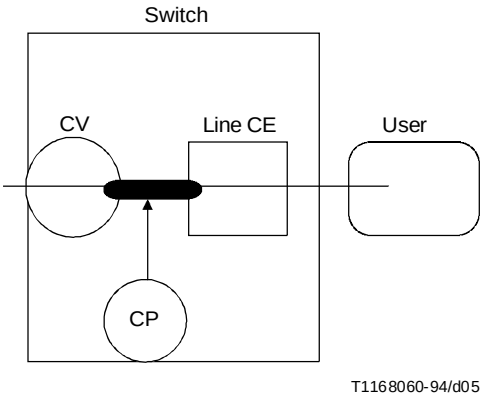


FIGURE 5/Q.1301

**Line CE**

The call related events are those events reported by transitions in the originating and terminating basic call view.

#### 8.1.2.2 Attributes

**Line Identifier** – Each Line CE has at least one unique identifier. Some Line CEs may have multiple identifiers.

**Call View Identifiers** – One or more call view identifiers may be associated with each Line CE.

**CP Identifiers** – One or more CP Identifiers may be associated with each Line CE. A CP identifier is comprised of a CE identifier, and CV identifier if required for unambiguous identification.

**User** – A user is associated with a Line CE.

**State** – The condition of the Line CE is described by Enable, Disable, In-service, Out-of-service, etc.

#### 8.1.2.3 Actions

The actions associated with a Line CE are those provided by management (e.g. Enable, Disable) and those provided by user action (e.g. Set-Forwarding, Clear Forwarding, etc.).

#### 8.1.2.4 Notifications

The notifications associated with a Line CE are those provided by management (e.g. Enabled, Disabled) and those provided by user action (e.g. Forwarding-Set, Forwarding-Clear, etc.).

### 8.1.3 Distribution CE

#### 8.1.3.1 Description and behaviour

An entity that distributes calls to other CEs.

#### 8.1.3.2 Types

**Incoming Distribution CE** – An entity that distributes calls to Line CEs and other incoming Distribution CEs.

**Outgoing Distribution CE** – Not defined by this Recommendation.

#### 8.1.3.3 Incoming Distribution CE

##### 8.1.3.3.1 Description and Behaviour

An entity that distributes calls to Line CEs or other Incoming Distribution CEs.

The call related events are those events reported by transitions in the Incoming Distribution CE.

##### 8.1.3.3.2 Attributes

**Incoming Distribution CE Identifier** – Each Incoming Distribution CE has at least one unique identifier. Some Incoming Distribution CEs may have multiple identifiers.

**Call View Identifiers** – One or more call view identifiers may be associated with each Incoming Distribution CE.

**CP Identifiers** – One or more CP Identifiers may be associated with each lncoming Distribution CE. A CP identifier is comprised of a CE identifier, and CV identifier if required for unambiguous identification.

**CE Identifiers** – One or more Line or Incoming Distribution CE Identifiers to which calls are to be distributed may be associated with an Incoming Distribution CE.

**State** – The condition of the Incoming Distribution CE is described by Enable, Disable, In-service, Out-of-service.

##### 8.1.3.3.3 Actions

The actions associated with a Incoming Distribution CE are those provided by management (e.g. Enable, Disable, Query).

#### 8.1.3.3.4 Notifications

The notifications associated with an Incoming Distribution CE are those provided by management (e.g. Enabled, Disabled, QueryResult).

## 8.2 Communication Party

### 8.2.1 Description and behaviour

When a CP is specified by the computer it should at least identify the CE. However, the omission of a Call View should not make the CP ambiguous to the switch (i.e. preventing the switch from identifying a single relationship).

When a CP is specified by the switch then both the Call View and the CE should be specified.

### 8.2.2 Types

**Line CP** – Maintains an abstract call view association between a Line CE and an Originating or Terminating Call View which is direction independent.

**Distribution CP** – Maintains an abstract call view association between a Distribution CE and a Call View which is direction independent.

### 8.2.3 Line CP

#### 8.2.3.1 Description and behaviour

A Line CP maintains an abstract call view association between a Line CE and a particular Call View which is direction independent. See Figure 6.

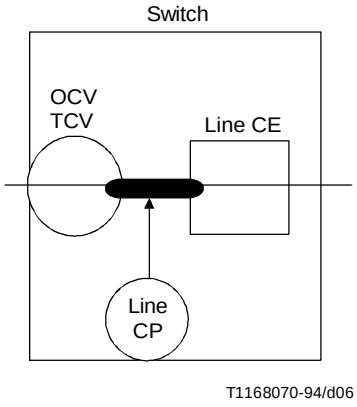


FIGURE 6/Q.1301

**Line CP**

#### 8.2.3.2 Attributes

**Line CP Identifier** – Each Line CP has a unique identifier. A CP identifier is comprised of a CE identifier, and CV identifier if required for unambiguous identification.

**Call View Identifier** – Only one Call View ldentifier may be associated with each Line CP.

**Line CE** – Only one Line CE may be associated with each Line CP.

**State** – The abstract state association between the Line CE and a Call View (Null, Active, Held).

#### 8.2.3.3 Actions

The actions associated with a Line CP are those provided by management (Enable, Disable, Query).

#### 8.2.3.4 Notifications

The notifications associated with a Line CP are those provided by management (Enabled, Disabled, QueryResult).

### 8.2.4 Distribution CP

#### 8.2.4.1 Description and behaviour

Distribution CP maintains an abstract call view direction independent association between a Distribution CE and a Distribution Call View.

#### 8.2.4.2 Types

**Incoming Distribution CP** – An entity that maintains an abstract call view association between an Incoming Distribution CE and a Distribution Call which is direction independent.

**Outgoing Distribution CP** – Not defined by this Recommendation.

#### 8.2.4.3 Incoming Distribution CP

##### 8.2.4.3.1 Description and behaviour

Incoming Distribution CP maintains an abstract call view association between an Incoming Distribution CE and an lncoming Distribution Call View which is direction independent. See Figure 7.

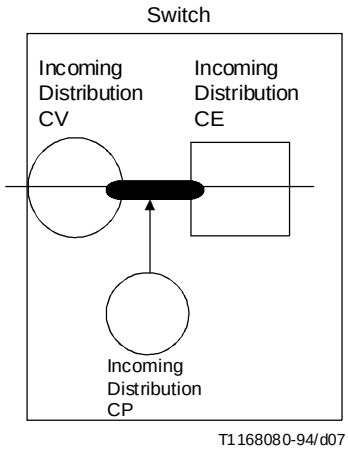


FIGURE 7/Q.1301

**Incoming Call Distribution CP**

#### 8.2.4.3.2 Attributes

**Incoming Distribution CP Identifier** – Each Incoming Distribution CP has a unique identifier. A CP identifier is comprised of a CE identifier, and CV identifier if required for unambiguous identification.

**Call View Identifier** – Only one Call View identifier may be associated with each Incoming Distribution CP.

**Incoming Distribution CE** – Only one Incoming Distribution CE may be associated with each Incoming Distribution CP.

**State** – The abstract state association between the incoming Distribution CE and an Incoming Distribution Call View (Null, Active, Held).

#### 8.2.4.3.3 Actions

The actions associated with an Incoming Call Distribution CP are those provided by management (Enable, Disable, Query).

#### 8.2.4.3.4 Notifications

The notifications associated with an Incoming Call Distribution CP are those provided by management (Enabled, Disabled, QueryResult).

## 8.3 User

### 8.3.1 Description and Behaviour

An entity that makes direct use of a CE (e.g. initiates or answers a Call). A user may be associated with multiple CEs and Call Views. See Figure 8.

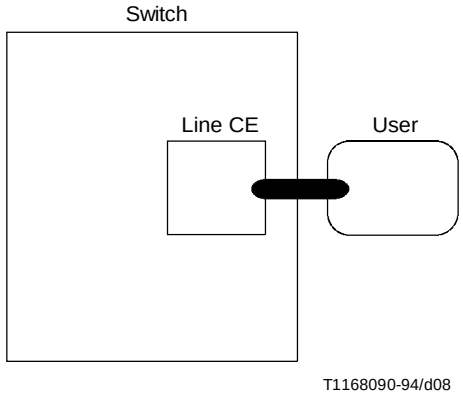


FIGURE 8/Q.1301

**User**

### 8.3.2 Types

**Registered User** – A user that is identified by some log-on process.

- **Agent** – A type of registered user that is distinguished from other users by being able to log-on to systems which distribute calls.
- **Other Registered User** – A type of registered user that performs a registration or explicit identification action, e.g. authorization code.

**Non-Registered User** – A user whose identity cannot be ascertained by TASC.

### 8.3.3 Registered User

#### 8.3.3.1 Description and behaviour

A user that is identified by some log-on process.

#### 8.3.3.2 Attributes

**Registered User Identifier** – Each registered user has a unique identifier.

**Call View Identifiers** – One or more Call View Identifiers may be associated a single each registered user.

**Line CEs** – One or more Line CE Identifiers may be associated with a single registered user.

**State** – The condition of the registered user at a Line CE (e.g. Null, Active).

#### 8.3.3.3 Actions

The actions associated with a User are those provided by management (Enable, Disable, Query).

#### 8.3.3.4 Notifications

The notifications associated with a User are those provided by management (Enabled, Disabled, QueryResult).

#### 8.3.3.5 Agent

##### 8.3.3.5.1 Description and behaviour

A type of registered user that is distinguished from other users by being able to log-on to systems which distribute calls. Agents may be members of one or more pools of agents or agent groups. An example of a system which coordinates and distributes calls is an automatic call distribution system. The agent group can be identified by an agent group identifier. Agents are represented in TASC as agent objects. In subsequent text, the word agent means agent object.

Agents have agent identifiers which uniquely identify the agent. Each agent may also have an agent password to be used during log-on for security.

Agents may control their availability to receive calls by invoking agent operations such as log-on, log-off, and indicating ready or not ready to accept calls. Agents have states and agent operations result in state changes. The agent state may be used by the call distribution system to determine the availability of agents. An agent may or may not be in different states in different groups.

Agent events may be reported by the switch to the computer when an agent invokes an operation at a telephone station or when an agent invokes one of the Manipulate Feature Functional Services (see Recommendation Q.1302).

##### 8.3.3.5.2 Attributes

**Agent Identifier** – Each Agent has a unique identifier.

**Password** – Each Agent may have a unique password.

**Agent Groups** – Agents may be members of one or more groups.

**Call View Identifiers** – One or more Call View Identifiers may be associated with each Agent.

**Line CE** – One or more Line CEs may be associated with each Agent.

**State** – The condition of an Agent at a Line CE is described by Null, Agent-Logged-On, Agent-Logged-Off, Agent-Ready, Agent-Busy, Agent-Working-After-Call.

#### 8.3.3.5.3 Actions

The actions associated with an agent are those provided by management (Enable, Disable, Query) and those provided by agent action (Agent-Log-in, Agent-Log-Out, Agent-Ready, Agent-Busy, Agent-Working-After-Call).

#### 8.3.3.5.4 Notifications

The notifications associated with an agent are those provided by management (Enabled, Disabled QueryResult) and those provided by agent action (Logged-in, Logged-Out, Ready, Not-Ready, QueryResult).

### 8.3.3.6 Other registered user

#### 8.3.3.6.1 Description and Behaviour

A type of registered user that performs a registration or explicit identification action, e.g. authorization code. This user will have access to different services and those services may be limited by the registration process. For example, GVNS or CUG.

#### 8.3.3.6.2 Attributes

**Other Registered User Identifier** – Each Other Registered User has a unique identifier.

**Password** – Each Other Registered User may have a unique password.

**Other Registered User Groups** – Other Registered User may be members of one or more groups.

**Call View Identifiers** – One or more Calls may be associated with an Other Registered User .

**Line CE** – One or more Line CEs may be associated with an Other Registered User.

**State** – The condition of the Other Registered User at a Line CE is described by (e.g. Null, Active).

#### 8.3.3.6.3 Actions

The actions associated with an Other Registered User are those provided by management (Enable, Disable, Query) and those provided by the Other Registered Users action (e.g. authorization code).

#### 8.3.3.6.4 Notifications

The notifications associated with an Other Registered User are those provided by management (Enabled, Disabled QueryResult) and those provided by Other Registered Users actions.

### 8.3.4 Non-Registered User

#### 8.3.4.1 Description and behaviour

A user whose identity cannot be ascertained by TASC.

#### 8.3.4.2 Attributes

**Non-Registered User Identifier** – A non-registered user may be recognized by the Line CE Identifier with which the user is associated.

**Call View Identifiers** – One or more Call View identifiers may be associated with each non-registered user.

**Line CE** – One Line CE identifier may be associated with each non-registered user.

**State** – The condition of the non-registered user at a Line CE (e.g. Null, Active, etc.).

#### 8.3.4.3 Actions

The actions associated with a Non-Registered User are those provided by management (Enable, Disable, Query).

#### 8.3.4.4 Notifications

The notifications associated with a Non-Registered User are those provided by management (Enabled, Disabled QueryResult).

## 8.4 Call View

There is a relationship among the CEs and Call Views. For a particular CE a particular Call View is presented. This Call View may present a unique set of events which is derived from the call related attributes of the CE. Each Call view presents a single-ended view of a call which is the view of call progress and call states from a switch's perspective for that particular CE.

The call view for a Line CE is separated into originating and terminating Call Views.

The following are the currently recognized Call Views.

### 8.4.1 Originating Call View

#### 8.4.1.1 Description and behaviour

A type of Call View that has one CE originating the call and one CP.

#### 8.4.1.2 Attributes

**Call View Identifier** – Each Originating Call View has a unique identifier.

**Line CE Identifier** – Each Originating Call View has one Line CE.

**Line CP Identifier** – Uniquely identifies the relationship between one CE and one call view (originating). A CP identifier is comprised of a CE identifier, and CV identifier if required for unambiguous identification.

**Users** – Each Originating Call View has one User.

**Call View State** – The condition of a Originating Call View.

1) NULL
2) PENDING
3) ORIGINATING
4) DELIVERED
5) ESTABLISHED
6) FAILED

#### 8.4.1.3 Actions

The actions associated with a call view are those provided by management (e.g. Query) and those actions initiated by application or user action (Make-Call, Answer-Call, Clear-Call, Hold-Call, Retrieve-Call, Transfer-Call, etc.).

#### 8.4.1.4 Notifications

The actions associated with a call view are those provided by management (e.g. QueryResult) and those actions as a result of application or user action (Service-Pending, Call-Delivered, Call- Established, Call-Cleared, etc.).

### 8.4.2 Terminating Call View

#### 8.4.2.1 Description and behaviour

A type of Call that has one terminating CE and one CP.

#### 8.4.2.2 Attributes

**Call View identifier** – Each Terminating Call View has a unique identifier.

**Line CE Identifier** – Each Terminating Call View has one Line CE.

**Line CP Identifier** – Uniquely identifies the relationship between one CE and one call view (terminating). A CP identifier is comprised of a CE identifier, and CV identifier if required for unambiguous identification.

**Users** – Each Terminating Call View has one User.

**Call View State** – The condition of a Terminating Call View.

1) NULL
2) ARRIVED
3) RECEIVED
4) ESTABLISHED
5) FAILED

#### 8.4.2.3 Actions

The actions associated with a call view are those provided by management (e.g. Query) and those actions initiated by an application or user action (Make-Call, Answer-Call, Clear-Call, Hold-Call, Retrieve-Call, Transfer-Call, etc.).

#### 8.4.2.4 Notifications

The actions associated with a call view are those provided by management (e.g. QueryResult) and those actions as a result of application or user action (Call Arrived, Call Received, Call Established, Call Cleared).

### 8.4.3 Incoming Distribution Call View

#### 8.4.3.1 Description and behaviour

A type of Call View that may have one Incoming Distribution CE and one Incoming Distribution CP. An incoming Distribution Call View describes the behaviour when an incoming call arrives at and is manipulated by a distribution function within a switch.

#### 8.4.3.2 Attributes

**Call View Identifier** – Each Incoming Distribution Call view has a unique identifier.

**Incoming Distribution CEs** – Each Incoming Distribution Call has at least one Line CE and may have two Line CEs.

**Incoming Distribution CPs** – Each Incoming Distribution Call has at least one Incoming Distribution CP.

**Incoming Distribution Call State** – The condition of an Incoming Distribution Call are the following states:

1) NULL
2) DISTRIBUTED
3) FAILED

#### 8.4.3.3 Actions

The actions associated with an Incoming distribution call view are those provided by management (e.g. Query).

#### 8.4.3.4 Notifications

The actions associated with an Incoming distribution call view are those provided by management and state transitions (e.g. QueryResult).

# 9 Call view states

The call view states are an abstraction of the call processing activity in the switch that particularly relates to a CE. Not all call processing activities are reflected in the TASC call view. For example, during the PENDING state, the destination CE address information is collected, yet the collection of the digit is not reported to the computer.

Call view state transitions are reflected by call progress events which are reported at the connection. These call progress events uniquely indicate the new call view state.

The following template is used to describe the call states:

**Description** – Describes the state.

**Entry Event Reported** – Identifies the event reported for the state transition.

## 9.1 Call View State descriptions

### 9.1.1 Originating states

Figure 9 shows the state transition diagram from the originating end perspective. The following originating states are defined:

1) NULL
2) PENDING
3) ORIGINATED
4) DELIVERED
5) ESTABLISHED
6) FAILED

#### 9.1.1.1 NULL state

**Description**

The state where no call exists from the perspective of the TASC single-ended view. For example, the switch may verify the authorization of this CE to place an outgoing call with given properties (e.g. bearer capability or CE restrictions) before exiting the NULL state. The type of authorization may vary for different types of originating resources.

**Entry Event Reported:** Call_Cleared.

#### 9.1.1.2 PENDING state

**Description**

The switch is collecting information for the purpose of processing the call. In addition, other information may be collected.

**Entry Event Reported:** Service_Pending.

#### 9.1.1.3 ORIGINATED state

**Description**

Destination address analysis, route selection, and routing initiation are performed by the switch. Authorization may be performed.

**Entry Event Reported:** Call_Originated.

#### 9.1.1.4 DELIVERED state

**Description**

The terminating end of the call is alerting.

**Entry Event Reported:** Call_Delivered.

#### 9.1.1.5 ESTABLISHED state

**Description**

The destination CE has answered the call and the parties in the call may exchange information.

**Entry Event Reported:** Call_Established.

#### 9.1.1.6 FAILED state

**Description**

Normal call progression has been aborted. Call failure indication is provided to calling party (e.g. busy).

**Entry Event Reported:** Call_Failed.

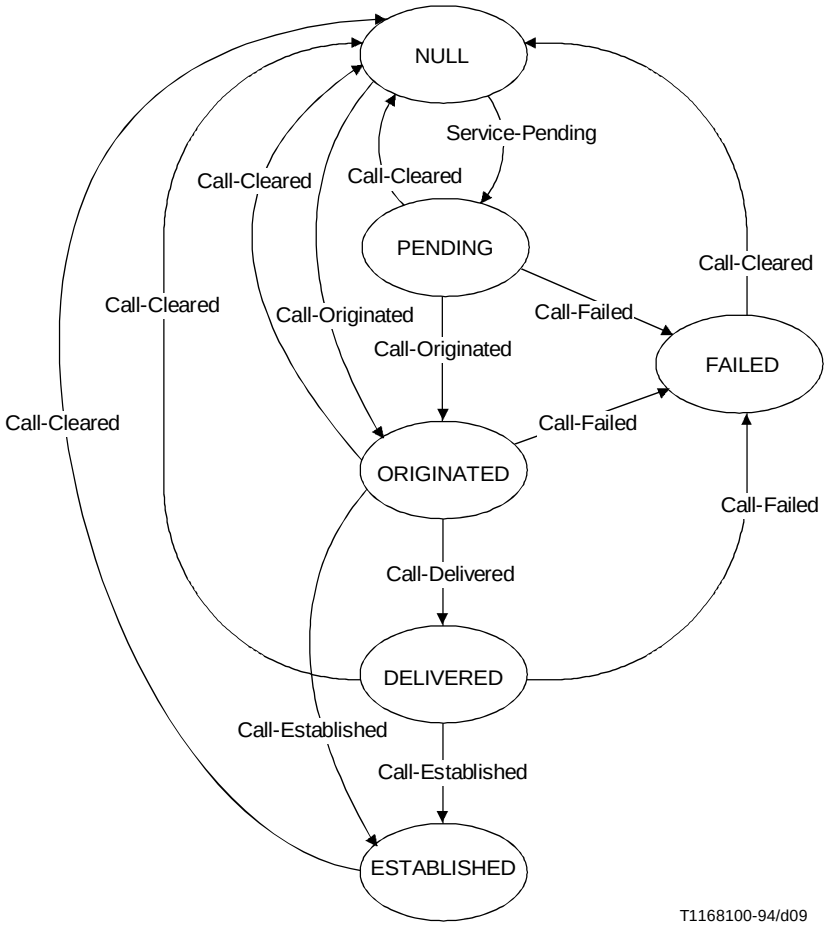


FIGURE 9/Q.1301

**Originating call view states**

### 9.1.2 Terminating Call View States

Figure 10 shows the state transition diagram from the terminating end perspective. The following terminating call view states are defined:

1) NULL
2) ARRIVED
3) RECEIVED
4) ESTABLISHED
5) FAILED

#### 9.1.2.1 NULL state

**Description**

The state where no call exists from the perspective of the TASC single-ended view. For example, the authorization to route an incoming call to the terminating CE may be performed (e.g. business group restrictions, restricted incoming access to a CE).

**Entry Event Reported:** Call_Cleared.

#### 9.1.2.2 ARRIVED state

**Description**

A call has arrived at or within the switch and the destination CE (e.g. telephone set, etc.) has been identified. No validation has been done as to the suitability of the selected CE for termination of the call (i.e. the CE may be busy, out of service, restricted, etc.).

**Entry Event Reported:** Call_Arrived.

#### 9.1.2.3 RECEIVED state

**Description**

The terminating end of the call is alerting.

**Entry Event Reported:** Call_Received.

#### 9.1.2.4 ESTABLISHED state

**Description**

The two CEs are connected. Information may be exchanged between the two users participating in the call.

**Entry Event Reported:** Call_Established.

#### 9.1.2.5 FAILED state

**Description**

Normal call progression has been aborted. Call failure indication is provided to calling party (e.g. busy).

**Entry Event Reported:** Call_Failed.

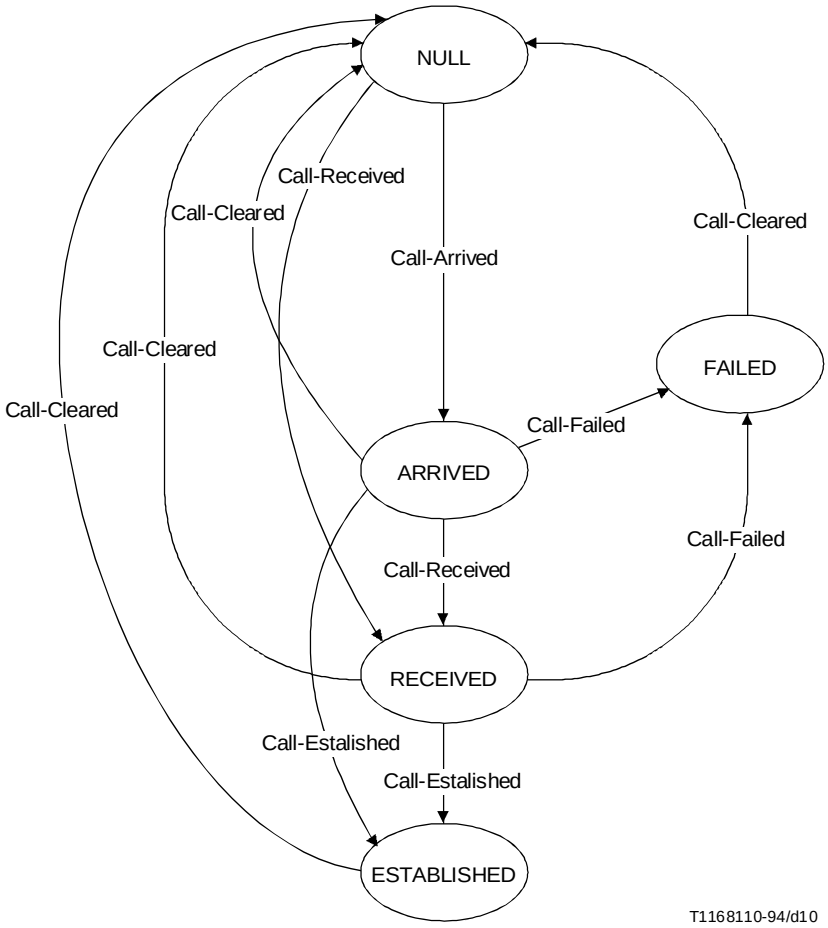


FIGURE 10/Q.1301

**Terminating call view states**

### 9.1.3 Incoming Distribution Call View

Figure 11 shows the state transition diagram from the incoming Distribution Call View perspective. The following distribution call view states are defined:

1) NULL
2) DISTRIBUTED ARRIVED
3) FAILED

#### 9.1.3.1 NULL

**Description**

The state where no call exists from the perspective of the TASC single-ended view.

**Entry Event Reported:** Call_Cleared (with a cause of distributed).

#### 9.1.3.2 DISTRIBUTED ARRIVED

**Description**

A call has arrived at or within the switch and the destination CE (e.g. ACD, hunt group) has been identified. No validation has been done as to the suitability of the selected CE for termination of the call (i.e. the CE may be busy, out-of-service, restricted, etc.).

**Entry Event Reported:** Call_Arrived (with a cause of distributed).

#### 9.1.3.3 FAILED

**Description**

Normal call progression has been aborted.

**Entry Event Reported:** Call_Failed (with a cause of distributed).

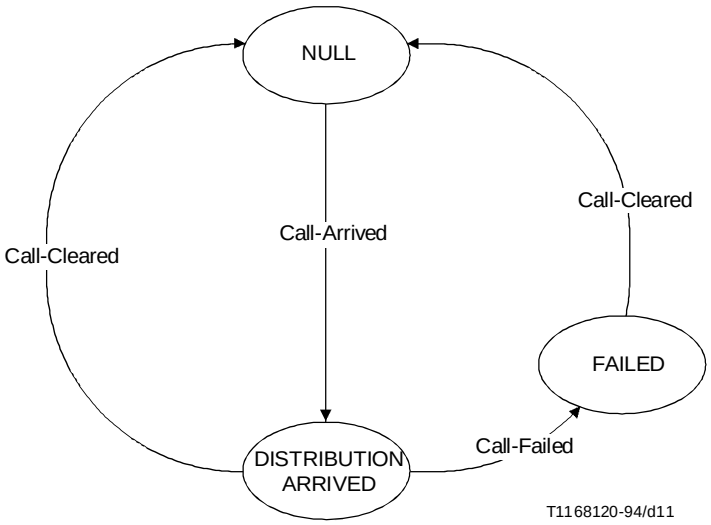


FIGURE 11/Q.1301

**Incoming Distribution Call View states**

#### 9.1.3.4 State diagrams

### 9.1.4 Agent model

The agent Model consists of the concepts of agent States and State Transitions.

#### 9.1.4.1 Agent states

The following states are defined for an agent:

| | |
|---|---|
| AGENT NULL | The state where an Agent is not logged on to a CE. Logging-on to a CE causes a transition from this state and Logging-off from a CE causes a transition to this state. |
| AGENT NOT READY | The state where an Agent is logged on to a CE but is not prepared to handle calls distributed by the distribution CE. An Agent may receive calls which are not handled by the distributor while in this state. |
| AGENT READY | The state where an Agent is logged-on to a CE and is prepared and waiting to handle calls from the distribution function. |
| AGENT BUSY | The state where a CE, on behalf of an Agent, participates in a call. |
| AGENT WORKING AFTER CALL | The state where a CE, on behalf of an agent, inactively participates in a call. While in this state the agent cannot receive further calls from the distributor function but may well be performing administrative duties for a previous call, e.g. updating a business order form. |

### 9.1.4.2 Agent diagram and transitions

Figure 12 shows the agent state model and agent state transitions.

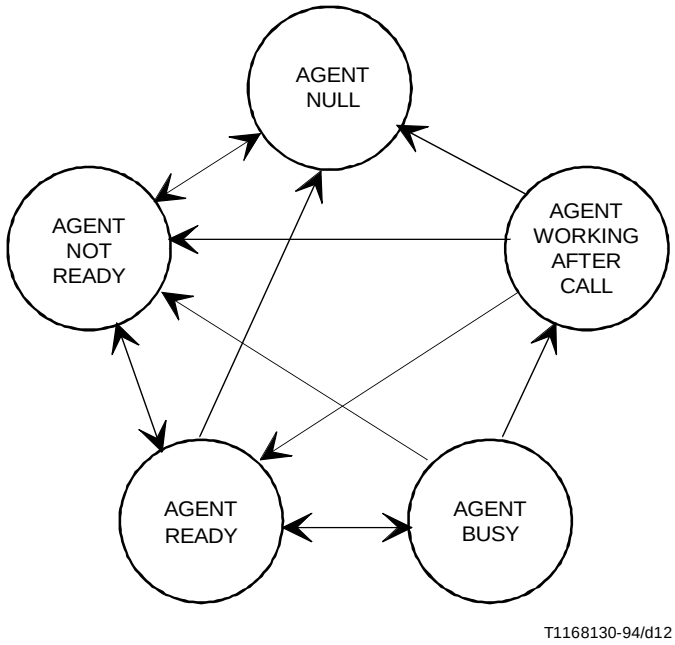


FIGURE 12/Q.1301

**Agent state model**

Agent state transitions are shown in the Table 1 below. The original state is shown in the far left-hand column and the state to which the agent is moving shown across the top. The valid events which will be reported are shown within the table body.

TABLE 1/Q.1301

**Agent state transitions**

| | AGENT NULL | AGENT NOT READY | AGENT READY | AGENT BUSY | AGENT WORKING AFTER CALL |
|---|---|---|---|---|---|
| AGENT NULL | | Agent_logged_on | | | |
| AGENT NOT READY | Agent_logged_off | | Agent_ready | | |
| AGENT READY | Agent_logged_off | Agent_not_ready | | Agent_busy | |
| AGENT BUSY | | Agent_not_ready | Agent_ready | | Agent_working_after_call |
| AGENT WORKING AFTER CALL | Agent_logged_off | Agent_not_ready | Agent_ready | | |

The agent events are:

- Agent_Logged_On
- Agent_Logged_Off
- Agent_Not_Ready
- Agent_Ready
- Agent_Busy
- Agent_Working_After_Call

An agent will have access to both a data terminal for communicating to the application and a telephony terminal for communicating to the switch. An agent may perform operations at both of those terminals. If an agent logs on by using the telephony terminal, then the computer application will be informed of state changes by the events identified above. If an agent logs on by using the data terminal, then state changes will be notified to the switch by use of the Manipulate Agent Functional Service as defined within the TASC Functional Service definitions.