Chapter **3** *FUNCTIONAL CALL MODELXE "FUNCTIONAL CALL MODEL"§*

This chapter describes concepts from the ECMA CSTA standard that are important for TSAPI application programming. The information presented here is summarized from the CSTA specifications. The complete specification is available from ECMA at the address given in Chapter 2.

Terminology

The following sections provide TSAPI definitions and acronyms. For clarity, the ECMA terms begin with capital letters throughout this section.

Definitions

ACD AgentXE "CD Agent"§:

A telephony user that is a member of an inbound or outbound Automatic Call Distribution (ACD) group. ACD Agents first sign on (Login) to an ACD groups and then the ACD will distribute calls to the agent.

Active CallXE "Active Call"§:

The call (at a station) that is connected (in a talking state) at that station. More specifically, the Connection (see below) for the Active Call is in the Connected State (also see below).

Alerting CallXE "Alerting Call"§:

A call that is ringing at a Device. More specifically, the Connection (see below) for an Alerting Call is in the Alerting State. When the Device is a telephone, the Alerting Call is ringing the telephone instrument.

ApplicationXE "Application"§:

A co-operative process distributed between a Switching Function (see below) and a Computing Function (also see below).

Application DomainXE "Application Domain"§:

The union of one Switching Sub-Domain (see below) and one Computing Sub-Domain (also see below).

Basic CallXE "Basic Call"§:

A Call (see below) between exactly two Devices (also see below).

Call (TSAPI programming objectXE "TSAPI programming object"§):

A Switching Function communications relationship. Typically, a Call is a communications relationship between two or more Devices. Note, however, during call set-up and release, there may be only one Device on the Call. A Call is a TSAPI programming object.

Call Identifierxe "TSAPI programming handle"§:

3-2   Functional Call Model

A TSAPI programming handle that identifies a Call

Complex CallXE "Complex Call"§:

A Call connecting more than two Devices.

Computing DomainXE "Computing Domain"§:

Those computers (and their Objects) accessible from a Switching Function. Where a switch has multiple CTI links to multiple computers, the Computing Domain is the union of all computers connected to the switch.

Computing FunctionXE "Computing Function"§:

A computer, or other resource in a Computing Sub-domain.

Computing Sub-DomainXE "Computing Sub-Domain"§:

Those computers (and their Objects) accessible from the Switching Function using a specific CTI link. Where a switch has multiple CTI links to multiple computers, the Computing Sub-Domain is a subset of the Computing Domain. Where the switch has a single CTI link, the Computing Sub-Domain is equivalent to the Computing Domain.

ConnectionXE "Connection"§ (TSAPI programming objectXE "TSAPI programming object"§):

A relationship between a Call and a Device. A Connection is in one of a number of states (alerting, held, connected, etc.). Note that when a Call connects (for example) three Devices, there are three Connections for the Call. Each Connection reflects the state of the Call at one of the Devices.

Connection IdentifierXE "Connection Identifier"§ (TSAPI programming handleXE "TSAPI programming handle"§):

A TSAPI programming handle that identifies a Connection. A Call Identifier and a Device Identifier comprise a TSAPI Connection Identifier.

DeviceXE "Device"§ (TSAPI programming objectXE "TSAPI programming object"§):

An Object which abstracts the interface between a user and the communications signaling in the Switching Function. A Device can be a single endpoint (such as a telephone), or multiple endpoints that form a group (ACD group or trunk group).

Device Identifier (TSAPI programming handle):

A TSAPI programming handle that identifies a Device.

Directory NumberXE "Directory Number"§:

The phone number for a Device. Directory Numbers are typically denote telephone station Devices, but ACD groups and other Devices may have Directory Numbers also.

DomainXE "Domain"§:

The union of a Switching Domain and a Computing Domain.

EventXE "Event"§:

A stimulus of interest to an Application that (typically) causes a change in the state of a Device object.

Event ReportXE "Event Report"§:

A message from a Switching Sub-Domain to a Computing Sub-Domain indicating that an Event has occurred.

Held CallXE "Held Call"§:

A call (at a station) that is held (in a hold state) at that station. More specifically, the Connection (see below) for a Held Call is in the Hold State (also see below).

Interconnection Service BoundaryXE "Interconnection Service Boundary"§:

An abstraction of the boundary between the Switching Domain and the Computing Domain. In practice, CTI links bridge the Interconnection Service Boundary.

ObjectXE "Object"§:

An abstraction of some system entity for modeling or programming purposes. TSAPI programming objects include Connections, Calls, and Devices. Each has a corresponding programming handle, or identifier.

PartyXE "Party"§:

A telephony user. A Party may be a human, application, or other resource
(such as a port on a voice response unit).

ServiceXE "Service"§:

The benefit provided by an Application to a User.

Service BoundaryXE "Service Boundary"§:

A specific CTI interface between a Computing Function and a Switching Function. All Service boundaries cross the Interconnection Service Boundary.

StateXE "State"§:

An object's current condition. Specifically, TSAPI Connections have an associated state.

Switching DomainXE "Switching Domain"§:

Those switches (and their Objects) accessible from a Computing Function. Where a computer has multiple CTI links to multiple switches, the Switching Domain is the union of all switches connected to the computer.

Switching FunctionXE "Switching Function"§:

A switch in a switching Sub-domain.

Switching Sub-DomainXE "Switching Sub-Domain"§:

Those switches (and their Objects) accessible from a Computing Function using a specific CTI link. Where a computer has multiple CTI links to multiple switches, the Switching Sub-Domain is a subset of the Switching Domain. Where the computer has a single CTI link, the Switching Sub-Domain is equivalent to the Switching Domain.

UserXE "User"§:

A person, process or piece of equipment that receives direct benefit (e.g. new feature, improved performance) from an Application's Services.

Acronyms

ACD   Automatic Call Distribution

CSTA Computer-Supported Telecommunications Applications

ID   Identifier

ISDN Integrated Services Digital Network

3-6   Functional Call Model

ArchitectureXE "Architecture"§

This section summarizes the functional architecture underlying ECMA CSTA and TSAPI. CSTA defines the interworking between Computing and Switching Functions in a way which is independent of their physical implementation. This section introduces the concepts of:

- distribution of Computing and Switching Functions

- TSAPI Service

- client/server model

- and TSAPI objects

An Application is a co-operative process distributed between a Switching Function (switch) and a Computing Function (computer). This section describes the interactions between them.

Distribution of ComputingXE "Distribution of Computing"§ and Switching Functions

XE "Switching Functions"§One (or several computers in a computing network) provide the Computing Functions and one (or several) switches provide the Switching Functions for a TSAPI Application. The TSAPI application appears to a User (human or machine) to be a single application, not as two separate functions on two separate networks (as it is, in fact, implemented).

Since the applications use distributed resources, communications must occur between the distributed entities. Figure 3-1 shows an abstract communications model. Note that each of the distributed functions is expanded into:

- a client application component that provides the TSAPI interactions

- a server communications component that exchanges messages

- networking support, or lower layer interconnection

µ §

Figure 3-1 - Model Showing the Relationship Between TSAPI Elements

XE "relationship between TSAPI Elements"§
Figure 3-1 shows that distributed Application components use TSAPI definitions to interact with their peers. TSAPI defines the Service descriptions and provides the service interface between the Application functions and the Server providing communications with the switch. TSAPI supports various switches, and as a consequence, some of the TSAPI elements are optional and their use is implementation dependent.

API ServicesXE "API Services"§

Unless otherwise qualified, the TSAPI definition uses the term 'Service' to refer to the benefit that an application server provides to a client application. TSAPI Services are decoupled from the specific CTI link connecting the switch with the application server. Since TSAPI is independent of the particular telephone terminal types, the Switching Function must determine how to support a given TSAPI request for its specific telephone types. For example, TSAPI does not specify how to provide the Make Call Service for analog or ISDN telephones. A Switching Function will use its existing service definitions to provide TSAPI Services on telephones where that service already exists.

TSAPI definitions do not embody the specific details of how the Switching Function accomplishes TSAPI Services. The Switching Function does provide an abstraction to the TSAPI Service requester, via Event Reports, of the steps taken to accomplish the Service.

3-8   Functional Call Model

Service and Objects

A server provides a service to a client consists of monitoring and controlling Switching Sub-Domain objects. TSAPI defines the client application interface for monitoring and controlling these objects.

Functions

A Domain XE "Domain "§is the union of a Switching Domain and a Computing Domain. In other words, it is those switching and computing objects that an application can access. Figure 3-2 shows an example of a Domain. A heavy line divides the Domain into Switching and Computing Domains. The Switching Domain consists of Switching Functions S1, S2, and S3. Similarly, the Computing Domain consists of Computing Functions C1, C2, and C3. Each Function can a resident application with a view of the Domain where it resides. Note that if multiple Functions provide an application with the same view, then the Functions are in the same Sub-Domain. TSAPI Applications (shown in Figure 3-2 as "Application Domains") are distributed across at least one Switching Sub-Domain and at least one Computing Sub-Domain.

Computing Domain　　　　Switching Domain

Figure 3-2 Domains XE "Domains "§and Sub-DomainsXE "
Sub-Domains"§

TSAPI Switching Sub-Domain Model

XE "Switching Sub-Domain Model"§The Switching Sub-Domain Model defines an abstract view of a Switching Function. TSAPI defines several Switching Sub-Domain Model Objects for use in Application programming, Call, and Connection.

TSAPI DeviceXE "Device"§

A TSAPI application can monitor and control Devices of various types (including telephones). However, a TSAPI application may not be able to monitor or control all Devices. In CSTA, a Device can refer to either a physical device (such as buttons, lines, trunks, and stations) or a logical device (such as groups of devices, pilot numbers, and ACDs). Devices have associated attributes, which allow applications to monitor and control them.

TSAPI Device attributes are:

1. **Device Type** - A Device has one of the following types:

   - *ACDXE "ACD"§* - An Automatic Call Distributor (ACD) is a Switching Function mechanism that distributes calls to ACD agents. An ACD (as opposed to ACD-group) consists only of the distribution mechanism and not the ACD agents (or their Devices) to which the mechanism can distribute calls.

   - *ACD groupXE "ACD group"§* - An Automatic Call Distributor (ACD) group is the mechanism that distributes calls within a Switching Function as well as the ACD agent Devices to which that mechanism distributes calls.

   - *ButtonXE "Button"§* - is an instance of a call manipulation point at an individual station. Simple analog stations often have no physical buttons but behave as if they had one. Some advanced stations can emulate several analog stations and often represent those stations with several buttons. In some situations it is desirable to identify a given button on a multi-button station. Note that a station with several line appearance buttons could have either the same telephone number or different telephone numbers assigned to those buttons.

- *Button groupXE "Button group"§* - is two or more instances of a Button at an individual station.

- *LineXE "Line"§* - is a communications interface to one or more stations typically associated with a directory number. In some situations it may be impossible to identify individual stations that share a line (a single directory number).

- *Line groupXE "Line group"§* - is a set of communications interfaces to one or more stations.

- *Operator* XE "Operator "§- also known as AttendantXE "Attendant"§, is a device that is used to interact with a party to assist in call setup or to provide other telecommunications service. This device is different from other devices in that it is often involved in setting up other calls, and is usually not part of the call after the call is connected.

- *Operator groupXE "Operator group"§* - two or more operator devices used interchangeably or addressed identically.

- *StationXE "Station"§* - is the traditional telephone device. A station is a physical unit of one or more buttons and one or more lines.

- *Station groupXE "Station group"§* - is two or more stations used interchangeably or addressed identically.

- *TrunkXE "Trunk"§* - a device that spans switching sub-domains. In order to monitor and control calls that cross switching sub-domains it may be desirable to address the point at which the call crosses the boundary. This point is generally a trunk or trunk group.

- ***Trunk groupXE "Trunk group"§*** - often, many trunks connect to the same place. These trunks are often placed in groups and accessed using a single identifier. In such a configuration the individual trunks are used interchangeably.

2. **Device ClassXE "Device Class"§** - different classes of TSAPI devices can be monitored and controlled differently. A Device must belong to one and may belong to more than one of the following classes:

   - ***DataXE "Data"§*** - a device that is used to make digital data calls (either circuit switched or packet switched). This class includes computer interfaces and G4 facsimile machines.

   - ***Image*** XE "Image "§- a device that is used to make digital data calls involving imaging, or high speed circuit switched data in general. This class includes video telephones and CODECs.

   - ***Other*** XE "Other "§- a type of device not covered by data, image, or voice.

   - ***Voice*** - a device that is used to make audio calls. This class includes all normal telephones, as well as computer modems and G3 facsimile machines.

3. **Device IdentifierXE "Device Identifier"§** - a TSAPI programming handle for a Device that allows an application to uniquely identify each device at the API. Devices are identified using static and/or dynamic identifiers:

   - ***Static Device Identifier*** XE "Static Device Identifier "§- A Static Device Identifier is stable over time. It remains constant and unique over calls. A Static Device Identifier is typically the dialed number for the Device known by both

the Computing and Switching Functions.

It is sometimes useful for the Switching Function to convert long phone number identifiers to another, usually shorter, static form for subsequent use in service interaction. An example of this would be the transformation of a Public Directory Number to a Private Directory Number.

This transformation allows service interactions to be independent of the identification mechanism and allows reduction in the amount of data exchanged. This transformed number is known as a Short Form Static Device Identifier.

Some Switching Functions allow the same dialed number to be assigned to Devices of different types. Thus, a TSAPI application may also need to use the Device Type to uniquely address the Device.

- *Dynamic Device Identifier* XE "Dynamic Device Identifier "§- A switching Function may not always make a Static Device Identifier available for every Device on a call. This may occur because a static identifier may not be available (there is no dialed number identifier for the device), or because a dialed number does not unambiguously refer to a single device (i.e. a group identifier). In these cases the Switching Function assigns a Dynamic Device Identifier as a handle for the Device for the duration of the call. Management of the Dynamic Device Identifier is discussed in section 5, Dynamic Identifier Management.

4. **Device StateXE "Device State"§** - is a list of the Connection States for all the calls which are associated with the Device. For information about Connection states see *TSAPI Connections* later in this chapter.

Call

TSAPI applications can monitor and control calls (including call establishment and release). In certain operations, such as conference and transfer, one Device in a Call is replaced with another Device or two Calls merged into a single Call. In these situations, the TSAPI Call object is maintained as long as the communications relationship remains across each operation (i.e. the call survives transfer, conference, and forwarding operations). TSAPI Call object attributes are:

1. **Call IdentifierXE "Call Identifier"§** - a Call Identifier is a TSAPI programming handle that the Switching Function assigns to each Call. The Call ID may or may not be unique among all calls within a Switching Sub-Domain, but coupled with a Device ID the pair will form a unique Connection ID within a Switching Sub-Domain. To allow reference to a nascent call, The switch will assign a Call ID before a call is fully established. For example, a switch will assign a Call ID to an incoming call when the called Device is Alerting (the assignment is done before the call is answered).

   Certain Services merging multiple calls into a single call.. Examples of such TSAPI Services are Transfer and Conference.
   During operations of Services that merge multiple calls, the call identifier may change, but the call continues as a TSAPI object. The management of the call identifier is described in section 5, Dynamic Identifier Management.

2. **Call state** XE "Call state "§- is a list of the Connection states for all the Devices that are a part of the Call.

   For simplicity, common call states for two-party calls have been a single descriptive name. For example, a two party call with a Connection State of "Connected" at one station and a Connection State of "Alerting" at the other has a Call State of

3-14 Functional Call Model

"Delivered". Figure 3-3 gives the mapping of descriptive names to Connection State lists for two-party calls. station are provided as a single value, whereas uncommon call states are provided as a list. For more information on Connection States, see the following section, *TSAPI Connections*. The *Call States* section of this chapter gives further information about Call States.

TSAPI ConnectionsXE "TSAPI Connections"§

A Connection is a relationship between a Call and a Device. A TSAPI application can monitor or control a Connection. For example TSAPI Services Hold Call, Reconnect Call, and Clear Call all control Connections. Connections are TSAPI programming objects with the following attributes:

1. **Connection IdentifierXE "Connection Identifier"§** – is a TSAPI handle that is made up of a Call Identifier and Device Identifier. For a call there are as many Connection identifiers as there are associated devices. Similarly, for a device there are as many Connection identifiers as there are associated calls. The Connection Identifier is unique within a Switching Sub-Domain and within a single TSAPI server. A TSAPI application cannot use a Connection Identifier until it has received the identifier from the Switching Function.

2. **Connection StateXE "Connection State"§** - is the state of a call (in a Connection ID) at the Device (in a Connection ID). The Connection state always refers to a single Call/Device relationship. Snapshot Services report Connection States for Calls and Devices. Monitors report Events, which are changes in Connection States for the monitored entity. Figure 3-3 shows a simple Connection state model. Note that since TSAPI is switch independent and since switch features vary from switch to switch (and therefore interact differently on different switches) there is no definitive TSAPI Connection State model to which all switches comply.

Figure 3-3 Connection State Model

XE "Connection State Model"§   The   transitions   between
        states, shown by arrows, form the basis for providing Event
        Reports.

The TSAPI states are defined as follows.

- *NullXE "Null"§* - the state where there is no relationship
  between the call and device.

- *InitiatedXE "Initiated"§* - the state where the device is
  requesting service. Usually this results in the creation of a call.
  Often this is thought of as the "dialing" state.

- *AlertingXE "Alerting"§* - the state where a device is alerting
  (ringing).

- *Connected* XE "Connected "§- the state where a device can
  communicate with other Devices on a call (cannot be a held
  call).
- *Held* - the call is "on hold" at the Device.

- *Queued* XE "Queued "§- the state where normal state
  progression has been stalled. For example, a call being
  processed by an ACD that is waiting for an ACD agent to
  become available is "queued".

- *Failed* XE "Failed "§- the state where normal state progression
  has been aborted. a "Failed" state can result because of failure
  to connect to the calling (originator) device, failure to connect
  the called (destination) device, failure to create the call, and
  other reasons.

Call Status Event ReportsXE "Call Status Event Reports"§

The Switching Sub-Domain model abstracts a view of call states and events that Devices control using telecommunications signaling systems. This abstract view is probably more detailed than most applications require, but it introduces a precise language for describing Event Reports, Call States, and Service functional descriptions. Connection state changes correspond to telecommunications signaling at a Device.

ISDN specifications model network access as a distributed state machine. ECMA CSTA borrows from this ISDN model. One part of this access state machine resides in the Device. There is another similar distributed access state machine which resides across the ISDN network at the egress device.

Thus, a call can be modeled as a collection of Connection state machines. Network signaling causes corresponding changes in the state machines across the network. When signaling occurs, a state change occurs at the affected Connection. Figure 3-4 shows this concept of communication between distributed state machines for the case of establishing a simple call. For informational purposes, the ISDN call states have also been shown.

Time		Device D1	Call C1	Device D2	
T1	Null	Null	No Event Report	Null	Null
T2	Setup	Initiated	Service Initiated	Null	Null
T3	Proceeding	Connected	Originated	Null	Null
T4	Delivered	Connected	Delivered	Alerting	Receive
T5	Connected	Connected	Established	Connected	Connected

3-18   Functional Call Model

ISDN Call State	TSAPI Connection States	TSAPI Event Reports	TSAPI Connection States	ISDN Call State

Figure 3-3 - Relationship of Call State Changes and Event Reports to Network Signaling.

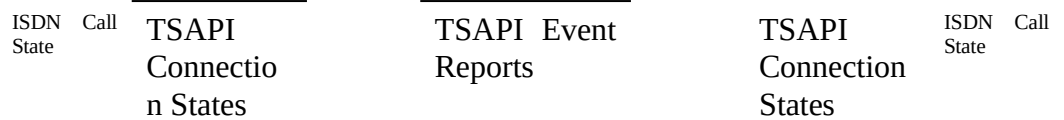XE " Relationship of Call State Changes and Event Reports to network"§Notice in Figure 3-3 that the State changes and Event Reports are based on Switching Function signaling. Many simple Connection events are of interest to applications. Certain telecommunications operations involve changes to many Connections. TSAPI reports these compound events (such as Transfer, Conference and Clear Call) in a single Event Report. Each TSAPI Event Report defines which Connections have changed state.

TSAPI Call StatesXE "TSAPI Call States"§

A Call State is defined as the list of Connection states for all the Devices involved in the call. This list is also called the **Compound Call StateXE "Compound Call State"§**. Listing the Connection states can describe any possible call state. However, most calls are often in a small number of widely recognized states.

TSAPI defines those states as the **Simple Call States** XE "Simple Call States"§shown in Table 3-1. TSAPI does not report Simple Calls States as a list, but rather in an abbreviated fashion. Note that Simple Call States can differ by the order of the Connection state list. Alerting-Connected is not equal to Connected-Alerting. The first is the Simple Call State "Received" and the second is "Delivered".

Null can be a known Connection state, so for a nascent call it is possible to have a Call state with only one non-Null Connection (see Table 3-1).

For calls with two Connections, the following table summarizes the Simple Call States assigned to the combinations of Connection States. If there is no entry in Table 3-1 for the combination of Connection states, then TSAPI provides the list.

For calls with more than two non-Null Connection states, The Call State is a compound call state. TSAPI (at times) simplifies the compound call state by relating it to a particular device. The Connection State related to a particular device in this way is called the **Local Connection StateXE "Local Connection State"§**. Other Connection States are not differentiated from one another. A three party conference call that is on hold at a given Device and connected to the other two devices has a Local Connection State of "Held" at that given Device.

Local Conn. State	Other Conn. State	Simple Call state
Alerting	Connected	Received
Alerting	Hold	Received-On Hold
Connected	Alerting	Delivered
Connected	Connected	Established
Connected	Failed	Failed
Connected	Hold	Established-On Hold
Connected	Null	Originated
Connected	Queued	Queued
Hold	Alerting	Delivered-Held

3-22   Functional Call Model

Hold	Connected	Established-Held
Hold	Failed	Failed-Held
Hold	Queued	Queued-Held
Initiated	Null	Pending
Null	Null	Null

Table 3-1 TSAPI Simple Call States

Dynamic Identifier Management

XE "Dynamic Identifier Management"§
Since Connection Identifiers comprise a Device ID and a Call ID, proper management of Connection Identifiers will, in turn, provide proper management of Dynamic Device Identifiers and Call Identifiers.

The Switching Function provides Connection Identifiers when either a new Call or Device Identifier is needed. When a call is made the switch provides a Connection Identifier. The switch then provides the Connection ID in any following Event Reports that pertain to that call. Similarly, the switch provides Connection IDs

containing a Device ID for a device involved in a call.

The switch updates identifiers when needed. If a Conference or Transfer (merging two calls) changes a Call ID, then the switch provides Event Reports containing Connection IDs that link the old call identifier to the new identifier. Similarly, if a Dynamic Device Identifier is changed, the switch -provides new Connection Identifiers for the devices in the call. Both Service Acknowledgments and Event Reports may contain information necessary to manage identifiers.

Identifiers cease to be valid when their context vanishes. If a call ends, its call identifier is no longer valid. Similarly, if a device is removed from service or from a call, its dynamic device identifier becomes invalid. Many Event Reports and Services specify when a Connection Identifier has lost or will lose its context.

Identifiers can be reused. Once an identifier has lost its context it may be re-used to identify another object. Most implementations will not reuse identifiers immediately.

Call and Device Identifiers can be, but are not guaranteed to be, globally unique. The TSAPI server ensures that the combination of Call and Device Identifier is globally unique within a switching sub-domain. To accomplish this, compliant PBX drivers ensure that either the call identifier, or the device identifier (or both) is globally unique. In many cases the Connection Identifier requires the use of both the Call and Device Identifiers to uniquely refer to Connections in a call.