
**Information technology —
Telecommunications and information
exchange between systems — Services
for Computer Supported
Telecommunications Applications (CSTA)
Phase III**

*Technologies de l'information — Télécommunications et échange
d'information entre systèmes — Services pour applications en
télécommunications supportées par ordinateur (CSTA) en phase III*

Reference number
ISO/IEC 18051:2000(E)



PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

© ISO/IEC 2000

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

Contents

Foreword	xiii
Introduction	xiv
1 Scope	1
2 Normative References	1
3 Conformance	2
3.1 Switching Function	2
3.1.1 Conformant Services	2
3.1.2 Conformant Events	3
3.1.3 CSTA Profiles	3
3.1.4 Support of Service Requests And Manual Mode	3
3.2 Special Resource Function Conformance	3
3.2.1 Conformant Services	3
3.2.2 Conformant Events	4
3.2.3 Support of Service Requests And Manual Mode	4
3.3 Computing Function Conformance	4
4 Definitions and Abbreviations	5
5 Functional Architecture	5
6 CSTA Operational Model	5
6.1 Switching Sub-Domain Model	6
6.1.1 Switching Sub-Domain Name	6
6.1.2 Application Working Domain	6
6.1.3 Device	6
6.1.4 Call	27
6.1.5 Connection	33
6.1.6 Call State Definitions	37
6.1.7 Referencing Devices, Elements, Appearances and Device Configurations	38
6.1.8 Management of Dynamically-Assigned Identifiers	40
6.2 Special Resource Functions	40
6.2.1 Voice Unit	40
6.3 I/O Services	42
6.3.1 Data Path Definition	42
6.3.2 I/O Registration Services	43
6.3.3 Data Path States and Operational Model	43
6.3.4 I/O Services Example	44
6.4 Call Detail Record (CDR) Services	44
6.4.1 CDR Services Examples	44
6.5 Capabilities Exchange	46

6.5.1	Switching Function Capabilities	46
6.5.2	Device Capabilities	47
6.5.3	Dynamic Feature Availability	47
6.6	Switching Function Information Synchronization	47
6.6.1	Switching Function Level Information	48
6.6.2	Device Level Information	48
6.6.3	Call Level Information	48
6.7	Status Reporting Services	48
6.7.1	System Status	48
6.7.2	Monitoring	50
6.7.3	Snapshot Services	53
6.8	Additional Services, Features & Behaviour	53
6.8.1	Forwarding	53
6.8.2	Connection Failure	55
6.8.3	Recall	57
6.8.4	Call Back	57
6.8.5	External Calls	58
6.8.6	Tracking a Diverted Call	59
6.8.7	Media Stream Access	59
6.8.8	Routeing Services	62
6.8.9	Device Maintenance	67
6.8.10	Prompting	67
6.8.11	Telephony Tones Features	67
6.8.12	DTMF and Rotary Pulse Digits Features	67
6.8.13	Data Collection Services	68
7	Association Establishment	68
7.1	Implicit Association	68
7.2	Explicit Association	69
8	Security Service	70
9	Generic Service Requirements	70
9.1	Service Request	70
9.2	Service Response (Acknowledgements)	71
9.2.1	Positive Acknowledgement Models	71
9.2.2	Negative Acknowledgement	72
9.3	Diagnostic Error Definitions	72
9.3.1	Error Categories	72
9.3.2	Error Values	72
9.4	Vendor Specific Extensions	73
9.4.1	Private Data	73
9.4.2	Escape Services and Private Event	74
9.5	General Services and Event Functional Requirements	75

9.5.1	Services	75
9.5.2	Events	76
10	CSTA Device Identifier Formats	77
10.1	Device Identifier Formats	77
10.1.1	Diallable Digits	77
10.1.2	Switching Function Representation	78
10.1.3	Device Number	80
10.2	Functional Requirements	80
11	Template Descriptions	80
11.1	Service Template	80
11.1.1	Service Description	80
11.1.2	Service Request	81
11.1.3	Service Response	81
11.1.4	Operational Model	82
11.2	Event Template	82
11.2.1	Event Description	82
11.2.2	Event Parameters	82
11.2.3	Event Causes	82
11.2.4	Functional Requirements	83
11.3	Parameter Type Template	83
11.3.1	Parameter Type Description	83
11.3.2	Format	83
11.3.3	Functional Requirements	83
12	Parameter Types	84
12.1	Definitions	84
12.2	Defined Parameter Types	85
12.2.1	AccountInfo	86
12.2.2	AgentPassword	86
12.2.3	AuthCode	86
12.2.4	CallCharacteristics	86
12.2.5	CallLinkageData	87
12.2.6	CallQualifyingData	88
12.2.7	ChargingInfo	89
12.2.8	ConnectionInformation	89
12.2.9	ConnectionList	90
12.2.10	CorrelatorData	90
12.2.11	CSTAPrivateData	91
12.2.12	CSTASecurityData	91
12.2.13	ErrorValue	92
12.2.14	EventCause	102
12.2.15	LocalConnectionState	104
12.2.16	MediaCallCharacteristics	105
12.2.17	MediaServiceType	106
12.2.18	MonitorFilter	107
12.2.19	ServicesPermitted	107

12.2.20	SimpleCallState	108
12.2.21	SystemStatus	109
12.2.22	TimeInfo	109
12.2.23	UserData	109
12.3	Identifier Parameter Types	111
12.3.1	AgentID	112
12.3.2	AssociatedCalledDeviceID	112
12.3.3	AssociatedCallingDeviceID	112
12.3.4	AuditoryApparatusID	113
12.3.5	ButtonID	113
12.3.6	CalledDeviceID	113
12.3.7	CallingDeviceID	114
12.3.8	CDRCrossRefID	114
12.3.9	ConnectionID	114
12.3.10	DCollCrossRefID	116
12.3.11	DeviceID	116
12.3.12	DisplayID	116
12.3.13	EscapeRegisterID	116
12.3.14	HookswitchID	116
12.3.15	IOCrossRefID	116
12.3.16	IORegisterReqID	117
12.3.17	LampID	117
12.3.18	MediaServiceInstanceID	117
12.3.19	MediaStreamID	117
12.3.20	MessageID	117
12.3.21	MonitorCrossRefID	118
12.3.22	NetworkCalledDeviceID	118
12.3.23	NetworkCallingDeviceID	118
12.3.24	RedirectionDeviceID	119
12.3.25	RingerID	119
12.3.26	RouteingCrossRefID	120
12.3.27	RouteRegisterReqID	120
12.3.28	ServiceCrossRefID	120
12.3.29	SubjectDeviceID	120
12.3.30	SysStatRegisterID	121
13	Capability Exchange Services	122
13.1	Services	122
13.1.1	Get Logical Device Information	123
13.1.2	Get Physical Device Information	132
13.1.3	Get Switching Function Capabilities	136
13.1.4	Get Switching Function Devices	151
13.1.5	Switching Function Devices	153
14	System Services	156
14.1	Registration Services	156
14.1.1	Change System Status Filter	157
14.1.2	System Register	159
14.1.3	System Register Abort	162
14.1.4	System Register Cancel	163
14.2	Services	164

14.2.1	Request System Status	165
14.2.2	System Status	167
14.2.3	Switching Function Capabilities Changed	169
14.2.4	Switching Function Devices Changed	170
15	Monitoring Services	171
15.1	Services	171
15.1.1	Change Monitor Filter	172
15.1.2	Monitor Start	174
15.1.3	Monitor Stop	178
16	Snapshot Services	179
16.1	Services	179
16.1.1	Snapshot Call	180
16.1.2	Snapshot Device	183
16.1.3	Snapshot CallData	186
16.1.4	Snapshot DeviceData	188
17	Call Control Services & Events	190
17.1	Services	190
17.1.1	Accept Call	191
17.1.2	Alternate Call	193
17.1.3	Answer Call	196
17.1.4	Call Back Call-Related	198
17.1.5	Call Back Message Call-Related	201
17.1.6	Camp On Call	204
17.1.7	Clear Call	206
17.1.8	Clear Connection	209
17.1.9	Conference Call	213
17.1.10	Consultation Call	216
17.1.11	Deflect Call	222
17.1.12	Dial Digits	225
17.1.13	Directed Pickup Call	228
17.1.14	Group Pickup Call	231
17.1.15	Hold Call	234
17.1.16	Intrude Call	236
17.1.17	Join Call	240
17.1.18	Make Call	244
17.1.19	Make Predictive Call	250
17.1.20	Park Call	255
17.1.21	Reconnect Call	258
17.1.22	Retrieve Call	260
17.1.23	Single Step Conference Call	262
17.1.24	Single Step Transfer Call	266
17.1.25	Transfer Call	269
17.2	Events	272
17.2.1	Bridged	273
17.2.2	Call Cleared	275
17.2.3	Conferenced	278
17.2.4	Connection Cleared	284
17.2.5	Delivered	288

17.2.6	Digits Dialed	292
17.2.7	Diverted	295
17.2.8	Established	299
17.2.9	Failed	303
17.2.10	Held	308
17.2.11	Network Capabilities Changed	310
17.2.12	Network Reached	313
17.2.13	Offered	317
17.2.14	Originated	322
17.2.15	Queued	325
17.2.16	Retrieved	329
17.2.17	Service Initiated	331
17.2.18	Transferred	334
18	Call Associated Features	338
18.1	Services	338
18.1.1	Associate Data	339
18.1.2	Cancel Telephony Tones	341
18.1.3	Generate Digits	343
18.1.4	Generate Telephony Tones	345
18.1.5	Send User Information	348
18.2	Events	350
18.2.1	Call Information	351
18.2.2	Charging	353
18.2.3	Digits Generated	354
18.2.4	Telephony Tones Generated	355
18.2.5	Service Completion Failure	358
19	Media Attachment Services & Events	361
19.1	Services	361
19.1.1	Attach Media Service	362
19.1.2	Detach Media Service	366
19.2	Events	369
19.2.1	Media Attached	370
19.2.2	Media Detached	371
20	Routeing Services	373
20.1	Registration Services	373
20.1.1	Route Register	374
20.1.2	Route Register Abort	376
20.1.3	Route Register Cancel	377
20.2	Services	378
20.2.1	Re-Route	379
20.2.2	Route End	380
20.2.3	Route Reject	382
20.2.4	Route Request	384
20.2.5	Route Select	386

20.2.6	Route Used	388
21	Physical Device Features	390
21.1	Services	390
21.1.1	Button Press	391
21.1.2	Get Auditory Apparatus Information	392
21.1.3	Get Button Information	394
21.1.4	Get Display	396
21.1.5	Get Hookswitch Status	398
21.1.6	Get Lamp Information	399
21.1.7	Get Lamp Mode	401
21.1.8	Get Message Waiting Indicator	404
21.1.9	Get Microphone Gain	405
21.1.10	Get Microphone Mute	406
21.1.11	Get Ringer Status	407
21.1.12	Get Speaker Mute	409
21.1.13	Get Speaker Volume	410
21.1.14	Set Button Information	411
21.1.15	Set Display	412
21.1.16	Set Hookswitch Status	414
21.1.17	Set Lamp Mode	415
21.1.18	Set Message Waiting Indicator	417
21.1.19	Set Microphone Gain	418
21.1.20	Set Microphone Mute	420
21.1.21	Set Ringer Status	421
21.1.22	Set Speaker Mute	423
21.1.23	Set Speaker Volume	424
21.2	Events	426
21.2.1	Button Information	427
21.2.2	Button Press	428
21.2.3	Display Updated	429
21.2.4	Hookswitch	431
21.2.5	Lamp Mode	432
21.2.6	Message Waiting	433
21.2.7	Microphone Gain	434
21.2.8	Microphone Mute	435
21.2.9	Ringer Status	436
21.2.10	Speaker Mute	437
21.2.11	Speaker Volume	438
22	Logical Device Features	439
22.1	Services	439
22.1.1	Call Back Non-Call-Related	440
22.1.2	Call Back Message Non-Call-Related	441
22.1.3	Cancel Call Back	443
22.1.4	Cancel Call Back Message	444
22.1.5	Get Agent State	445
22.1.6	Get Auto Answer	447
22.1.7	Get Auto Work Mode	448
22.1.8	Get Caller ID Status	449
22.1.9	Get Do Not Disturb	450
22.1.10	Get Forwarding	452

22.1.11	Get Last Number Dialed	455
22.1.12	Get Routeing Mode	456
22.1.13	Set Agent State	457
22.1.14	Set Auto Answer	461
22.1.15	Set Auto Work Mode	463
22.1.16	Set Caller ID Status	465
22.1.17	Set Do Not Disturb	466
22.1.18	Set Forwarding	468
22.1.19	Set Routeing Mode	470
22.2	Events	471
22.2.1	Agent Busy	472
22.2.2	Agent Logged Off	473
22.2.3	Agent Logged On	474
22.2.4	Agent Not Ready	475
22.2.5	Agent Ready	477
22.2.6	Agent Working After Call	478
22.2.7	Auto Answer	480
22.2.8	Auto Work Mode	481
22.2.9	Call Back	482
22.2.10	Call Back Message	483
22.2.11	Caller ID Status	484
22.2.12	Do Not Disturb	485
22.2.13	Forwarding	486
22.2.14	Routeing Mode	488
23	Device Maintenance Events	489
23.1	Events	489
23.1.1	Back In Service	490
23.1.2	Device Capabilities Changed	491
23.1.3	Out Of Service	492
24	I/O Services	493
24.1	Registration Services	493
24.1.1	I/O Register	494
24.1.2	I/O Register Abort	496
24.1.3	I/O Register Cancel	497
24.2	I/O Services	498
24.2.1	Data Path Resumed	499
24.2.2	Data Path Suspended	500
24.2.3	Fast Data	501
24.2.4	Resume Data Path	503
24.2.5	Send Broadcast Data	504
24.2.6	Send Data	506
24.2.7	Send Multicast Data	508
24.2.8	Start Data Path	510
24.2.9	Stop Data Path	512
24.2.10	Suspend Data Path	513
25	Data Collection Services	514
25.1	Services	514

25.1.1	Data Collected	515
25.1.2	Data Collection Resumed	518
25.1.3	Data Collection Suspended	519
25.1.4	Resume Data Collection	520
25.1.5	Start Data Collection	521
25.1.6	Stop Data Collection	523
25.1.7	Suspend Data Collection	524
26	Voice Unit Services & Events	525
26.1	Services	525
26.1.1	Concatenate Message	526
26.1.2	Delete Message	527
26.1.3	Play Message	528
26.1.4	Query Voice Attribute	530
26.1.5	Record Message	532
26.1.6	Reposition	534
26.1.7	Resume	536
26.1.8	Review	537
26.1.9	Set Voice Attribute	539
26.1.10	Stop	541
26.1.11	Suspend	542
26.1.12	Synthesize Message	544
26.2	Events	545
26.2.1	Play	546
26.2.2	Record	547
26.2.3	Review	548
26.2.4	Stop	549
26.2.5	Suspend Play	550
26.2.6	Suspend Record	551
26.2.7	Voice Attribute Changed	552
27	Call Detail Record (CDR) Services	553
27.1	Services	553
27.1.1	Call Detail Records Notification	554
27.1.2	Call Detail Records Report	555
27.1.3	Send Stored Call Detail Records	559
27.1.4	Start Call Detail Records Transmission	561
27.1.5	Stop Call Detail Records Transmission	563
28	Vendor Specific Extensions Services & Events	565
28.1	Registration Services	565
28.1.1	Escape Register	566
28.1.2	Escape Register Abort	567
28.1.3	Escape Register Cancel	568
28.2	Services	569
28.2.1	Escape	570
28.2.2	Private Data Version Selection	571
28.3	Events	572

28.3.1	Private Event	573
Annex A	- Device Appearances	574
Annex B	- ISDN User-User Information Element Encoding for CSTA	582
Annex C	- Capability Bitmap Parameter Types	584
Annex D	- Connection State Transition Examples	635

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 18051 was prepared by ECMA (as ECMA-269) and was adopted, under a special "fast-track procedure", by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by national bodies of ISO and IEC.

Annexes A, B, and C form a normative part of this International Standard. Annex D is for information only.

Introduction

This International Standard defines Phase III of Services for Computer Supported Telecommunications Applications (CSTA) for OSI Layer 7 communication between a computing network and a telecommunications network. This International Standard is part of a Suite of Standards and Technical Reports for Phase III of CSTA. All of the Standards and Technical Reports in the Suite are based on practical experience of ECMA member companies and each one represents a pragmatic and widely-based consensus.

The evolution of this Suite began with CSTA Phase I, which included the CSTA Services and Protocol Standards (ECMA-179 and ECMA-180). In Phase II, Technical Report ECMA TR/68 was added illustrating how CSTA services and events may be used in typical call scenarios.

Phase III of CSTA extends the previous Phase II Standards (ECMA-217 and ECMA-218) in major theme directions as well as numerous details. This incorporates technology based upon the *versit* CTI Encyclopedia (Version 1.0), which was contributed to ECMA by *versit*. Major areas of advancement include:

- New categories of services and events such as capabilities exchange, charging, media attachment services, call data recording (CDR), etc.
- Additional services and events for call and device control.
- Enhancement to existing services and events.
- Organization of services and events to reflect a grouping based on function (call control, device control, etc.).
- Use of a consistent template for services and events that includes initial/final connection state, connection state transitions, event monitoring sequences, etc.

Information technology - Telecommunications and information exchange between systems - Services for Computer Supported Telecommunications Applications (CSTA) Phase III

1 Scope

This International Standard specifies the Services and Event Reports for Computer-Supported Telecommunications Applications, Phase III (CSTA).

This International Standard is focused on providing application service interfaces to a Switching Function, Computing Function and a Special Resource Function. A CSTA application interface is disassociated from the various user-network interfaces and network-network interfaces CSTA applications may serve, observe or manipulate. Because CSTA operates with existing telecommunications interfaces indirectly, it operates generically, so that differences among various existing interfaces are hidden from CSTA applications. Support of user-to-network interfaces is outside the scope of CSTA.

Although most terminal equipment (TE) are suitable for use with CSTA there will be instances of TE that will not be suitable in certain circumstances. Examples are:

- FAX terminals and modems that are unable to adjust their transmission modes to prevent carrier conflict when both parties are alerted via CSTA during call establishment;
- Functional terminals that perform telecommunication functions outside the control of the Switching Function.

Services defined in this International Standard allow functional integration between a computing network and a telecommunications network. Computing platforms (i.e., Application Programming Interfaces - APIs) that support such functionally-integrated applications are outside the scope of this International Standard.

Communication between the computing and switching (i.e., telecommunications) networks may take place via intervening networks ranging from simple point-to-point connections to local- or wide-area telecommunications networks.

This International Standard is part of a suite of CSTA Standards and Technical Reports that provide a comprehensive description of the architectural and practical issues involved in applying, implementing, and utilizing CSTA-based CTI applications.

2 Normative References

The following normative documents contain provisions which, through reference in this text, constitute provisions of this International Standard. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 8649:1996,	<i>Information technology - Open Systems Interconnection - Service definition for the Association Control Service Element</i> (this corresponds to ITU-T Rec. X.217 1995).
ISO/IEC 9545:1994,	<i>Information technology - Open Systems Interconnection - Application Layer structure.</i>
ISO/IEC 11571:1998,	<i>Information technology - Telecommunications and information exchange between systems - Private Integrated Services Network - Addressing.</i>

ISO/IEC 11572:2000,	<i>Information technology - Telecommunications and information exchange between systems - Private Integrated Services Network - Circuit mode bearer services - Inter-exchange signalling procedures and protocol.</i>
ISO/IEC 13712-1:1995,	<i>Information technology - Remote Operations: Concepts, model, and notation (this corresponds to ITU-T Rec. X.880, 1994).</i>
ISO/IEC TR 18053:2000,	<i>Information technology - Telecommunications and information exchange between systems - Glossary of definitions and terminology for Computer Supported Telecommunications Applications (CSTA) Phase III.</i>
ITU-T Rec. E.131:1988,	<i>Subscriber Control Procedures for Supplementary Telephone Services.</i>
ITU-T Rec. E.161:1995,	<i>Arrangement of Digits, Letters and Symbols on Telephones and Other Devices that can be used for Gaining Access to a Telephone Network.</i>
ITU-T Rec. E.164:1997,	<i>The International Public Telecommunication Numbering Plan.</i>
ITU-T Rec. H.225:1998,	<i>Call Signalling Protocols and Media Stream Packetization for Packet-based Multimedia Communications Systems.</i>
ITU-T Rec. Q.931:1998,	<i>ISDN User-Network Interface layer 3 Specification for Basic Call Control.</i>

3 Conformance

This Clause specifies the conformance requirements for a Switching Function, Special Resource Function, and a Computing Function.

Conformance requirements specify the parts of this International Standard that a CSTA conformant implementation shall support.

This International Standard specifies an operational model (Clause 6, “CSTA Operational Model” and Clause 9, “Generic Service Requirements”) that defines a collection of objects (e.g. domains and sub-domains, logical and physical elements, calls) and the relationships between these objects.

The behaviours of CSTA-conformant services, features, and event reports are determined by this model.

3.1 Switching Function

In order to conform to this International Standard a switching function shall support the following as a minimum:

1. the requirements pertaining to CSTA features as specified in Clause 6, “CSTA Operational Model”.
2. the requirements as specified in Clause 9, “Generic Service Requirements”.
3. the Get Switching Function Capability service.
4. either the Implicit or Explicit Application Association Establishment sequence, including the mandatory aspects in the sequence (such as the System Status service), as specified in Figure 7-1 and Figure 7-2.
5. at least one of the profiles as specified in 3.1.3, “CSTA Profiles”.

3.1.1 Conformant Services

In order to conform to a specific CSTA *service* an implementation shall support the following as a minimum:

1. the requirements of the service as specified by its service description, service request parameters, service response parameters, and operational model including connection state transitions, monitoring event sequences, and functional requirements.
2. the requirements associated with each parameter used in the service as specified by its parameter description, format, and functional requirements in Clause 12, “Parameter Types”.
3. all of the events that are associated with its service completion criteria, as documented in its event monitoring tables.
4. service requests that contain a device identifier parameter, an implementation shall support, at a minimum, the Diallable Digits format as specified in 10.1.1, “Diallable Digits”.

3.1.2 Conformant Events

In order to conform to a specific CSTA *event* an implementation shall support the following as a minimum:

1. the requirements of the event as specified by its event description, event parameters, event causes, and functional requirements.
2. the requirements associated with each parameter used in the event as specified by its parameter description, format, and functional requirements in Clause 12, "Parameter Types".
3. events that contain a device identifier parameter, an implementation shall support, at a minimum, the Switching Function Representation format as specified in 10.1.2, "Switching Function Representation".

3.1.3 CSTA Profiles

Some CSTA services and events are grouped together as profiles.

3.1.3.1 Basic Telephony Profile

This profile includes the following:

1. CSTA Services: Answer Call, Clear Connection, Make Call, Monitor Start (with the monitorType of device-type), and Monitor Stop.
2. CSTA Events: Connection Cleared, Delivered, Established, Failed, Network Reached, Originated, and Service Initiated.

Other CSTA services and events may be provided in any combination in addition to this set.

3.1.3.2 Routing Profile

If the switching function supports Routeing Services as specified in Clause 20, "Routeing Services", it shall support a minimum set of Routeing Services that includes: Route Request, Route Select, and Route End (from the switching function only).

Other Routeing services may be provided in any combination in addition to this set.

If a switching function supports the routeing for digital data calls, then the Route Register and CSTA Route Register Cancel shall also be included in the minimum set.

3.1.4 Support of Service Requests And Manual Mode

A conformant switching function may support a given service defined in this International Standard through the CSTA service boundary but is not required to support the equivalent service in a manual mode.

A conformant switching function may support a feature associated with an equivalent CSTA service defined in this International Standard through manual mode but is not required to support the equivalent service through the service boundary.

3.2 Special Resource Function Conformance

In order to conform to this International Standard a special resource function shall support the following as a minimum:

1. the requirements pertaining to CSTA features as specified in Clause 6, "CSTA Operational Model".
2. the requirements as specified in Clause 9, "Generic Service Requirements".
3. for a supported service, the special resource function shall not reject as unsupported all of the events specified in the monitoring event sequences associated with the service.
4. the atomic service request acknowledgment model as specified in 9.2, "Service Response (Acknowledgements)".

3.2.1 Conformant Services

In order to conform to a specific CSTA *service* an implementation shall support the following as a minimum:

1. the requirements of the service as specified by its service description, service request parameters, service response parameters, and operational model including connection state transitions, monitoring event sequences, and functional requirements.

2. the requirements associated with each parameter used in the service as specified by its parameter description, format, and functional requirements in Clause 12, "Parameter Types".
3. all of the events that are associated with its service completion criteria, as documented in its event monitoring tables.

3.2.2 Conformant Events

In order to conform to a specific CSTA *event* an implementation shall support the following as a minimum:

1. the requirements of the event as specified by its event description, event parameters, event causes, and functional requirements.
2. the requirements associated with each parameter used in the event as specified by its parameter description, format, and functional requirements in Clause 12, "Parameter Types".

3.2.3 Support of Service Requests And Manual Mode

A conformant special resource function may support a given service defined in this International Standard through the CSTA service boundary but is not required to support the equivalent service in a manual mode.

A conformant special resource function may support a feature associated with an equivalent CSTA service defined in this International Standard through manual mode but is not required to support the equivalent service through the service boundary.

3.3 Computing Function Conformance

In order to conform to this International Standard a computing function shall support the following as a minimum:

1. the requirements pertaining to CSTA features as specified in Clause 6, "CSTA Operational Model".
2. the requirements as specified in Clause 9, "Generic Service Requirements".
3. for a supported service, the computing function shall not reject as unsupported all of the events specified in the monitoring event sequences associated with the service.
4. the "Single Physical and Logical Element" and "Logical Element Only" device configurations as specified in 6.1.3.3, "Device Configurations".
5. for service requests, the Diableable Digits format of Device Identifiers as specified in 10.1.1, "Diableable Digits".
6. for events, all formats of Device Identifiers as specified in 10.1, "Device Identifier Formats".
7. the "No Appearance Addressability" and the "Individual Appearance Addressability" of referencing device elements as specified in 6.1.7, "Referencing Devices, Elements, Appearances and Device Configurations".
8. both types of service request acknowledgment models (e.g., Atomic and Multi-Step) as specified in 9.2, "Service Response (Acknowledgements)".
9. all failure models as specified in 6.8.2, "Connection Failure".
10. both switching function options of handling unsupported parameters in service requests as specified in the capability exchange services.
11. both the fixed and local view of the primaryOldCall and the secondaryOldCall parameters in the Conferenced and the Transferred events.
12. all bi-directional services for which it registered, whether explicitly (i.e., via a service registration service such as System Status Register) or implicitly (i.e., the switching function does not support registration but does support (as indicated through the capabilities exchange services) a particular bi-directional service and therefore may issue a service request to the computing function).
13. both the Implicit and Explicit Application Association Establishment sequences as specified in Clause 7, "Association Establishment".

4 Definitions and Abbreviations

The definitions and abbreviations used in this International Standard are defined in ISO/IEC TR 18053.

5 Functional Architecture

The objective of CSTA Architecture is to define the inter working mechanisms among Computing, Switching and Special Resource Functions independently from their physical implementations.

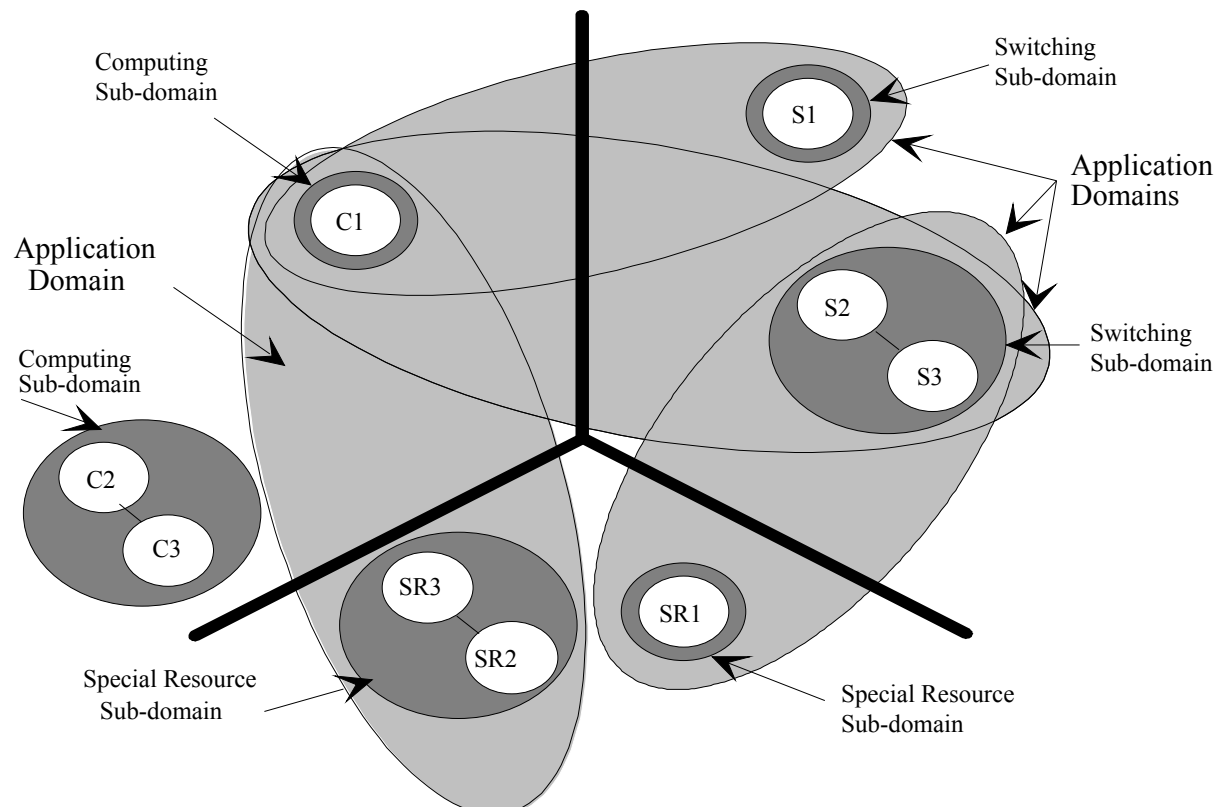
The concepts of: distribution of Computing, Switching and Special Resource Functions, CSTA Service, client server model, and CSTA objects as abstracted at a CSTA Service Boundary, are specified in another part of the CSTA Phase III Suite.

6 CSTA Operational Model

The operational model considered for CSTA is summarized in this clause.

The set of accessible Computing, Switching and Special Resource Functions from which an application might receive service defines a CSTA domain. An example of a CSTA domain is shown in the next figure. The CSTA domain contains switching, computing and special resource domains that are divided in the figure by the heavy lines. The special resource, switching and computing domains comprise Computing Functions (C1, C2 and C3), Switching Functions (S1, S2 and S3), and Special Resource Functions (SR1, SR2 and SR3). Each function can provide to a CSTA application, a view of the domain in which the function resides. Each such view defines a sub-domain. If one or more functions provides an identical view, then these functions are part of the same sub-domain. CSTA applications encompass at least two different sub-domains, and are represented in the next figure as application domains.

Figure 6-1 Domains and Sub-Domains



Note that a function may provide a view to an application that includes not only the objects within its sub-domain, but also the objects it can view in another (presumably related) sub-domain. For example (in Figure 6-1), a computing domain {C1} may receive a view of a switching sub-domain from a switching domain {S2+S3}. That

switching sub-domain may receive a view of a special resource sub-domain from a special resource domain {SR1}, and relay that view, in addition to the view of its sub-domain, to the Computing Function. This relay may preserve two views of separate switching and special resource sub-domains, or it may provide a combined view of a switching/special-resource sub-domain. As shown in the figure, {C1} also may have its own, direct view of a special resource sub-domain {SR2+SR3}. Finally, {C2+C3} represent a computing domain that is potentially, but not yet, involved in CSTA transactions with other sub-domains because an association has not yet been established between any other sub-domain and {C2+C3}.

6.1 Switching Sub-Domain Model

The tools needed to provide an abstract view of the Switching Function are defined by the switching sub-domain model. This model allows an application to conceptualize the Switching Function's operation. To provide this abstract view, CSTA defines several CSTA switching sub-domain model Objects that can be observed and acted upon by the Switching Function on behalf of the Computing Function. Those objects include CSTA Devices, Calls, and Connections.

6.1.1 Switching Sub-Domain Name

The switching function is identified by a unique name within the switching domain known at association time. This name may be used by the computing function to identify that different CSTA applications resulting from different associations are operating in the same switching sub-domain.

6.1.2 Application Working Domain

The application working domain is the subset of devices (and the calls and connections associated with those devices) inside a switching sub-domain that are controllable and/or monitorable over a CSTA Service Boundary. This subset is known at association time. The scope of this working domain may result from considerations like the application's design, licensing policy, security constraints, etc., and is under administration of the switching sub-domain. Different CSTA applications operating in the same switching sub-domain can have different working domains or share totally or partially the same working domain.

6.1.3 Device

CSTA enables manipulation and observation of devices that allow users to access telecommunications services.

Note 1 - It is not claimed that this International Standard alone supports ISDN (or any other) devices because, for example, of the additional information required to support such devices in PISNs. CSTA only provides a facility for passing ISDN (or other) specific information to allow, for example, a selection among ISDN devices sharing the same directory number (bearer capability, subaddress, etc.). Another example, that applies generally to telecommunications networks (including ISDN and OSI), is specifying the originator for a call that is established via CSTA. With the current signalling support, each party in a call can act only as a called party because the "network" is acting to originate the call. This situation has implications for both the network-to-terminal signalling and any application-level signalling that is significant to the calling party (e.g., issuing A_Associate).

Devices that are visible or controllable via CSTA are known as *CSTA Devices*.

CSTA Devices can be either physical devices (such as buttons, lines, trunks, and stations) or logical devices (such as groups of devices, pilot numbers, and automatic call distribution groups). CSTA Devices have attributes that allow CSTA to monitor and manipulate them. The attributes of any CSTA Device shall be:

1. **Device Type** - differing types of CSTA Device can be used for various purposes and can be manipulated and observed differently within CSTA. CSTA Device Types are listed and defined in 6.1.3.4, "Device Categories".
2. **Media Characteristics** - CSTA devices have distinct capabilities and characteristics defined by their media features. CSTA represents these characteristics by the following attributes.
 - **Media Class:** A CSTA Device shall belong to at least one and may belong to more than one media class. The Media Class can be used in Call Control services to help select a device for a call or it can be used in call control events to report the media class associated with the call. The following media classes are defined in CSTA:
 - **Audio** - 3.1 KHz audio. Devices in this class are used to make audio calls excluding speech calls. It includes G3 FAX and facsimile machines.
 - **Data** - Devices in this class are used to make digital data calls (both circuit switched and packet switched). This class includes digital computer interfaces and G4 facsimile machines.

- **Image** - Devices in this class are used to make digital data calls involving imaging, or high-speed, circuit-switched data in general. This class includes digital video telephones and CODECs.
- **Voice** - Devices in this class are used to make speech calls. This class includes standard telephones.
- **Other** - A class comprising devices not in the Data, Image, Audio or Voice classes.
- **Media Stream Information:** The media stream associated with a CSTA Device has attributes such as **Connection Rate**, **Bit Rate**, and **Delay Tolerance**. This information can be used in CSTA services to help select the media stream information for a call or to report the media stream information associated with an existing call.
- **Protocol Specific Information:** Many protocols provide additional information beyond what is standardized in CSTA to help distinguish devices. CSTA provides a mechanism where protocol specific information can be passed in CSTA messages to help select a specific device for a call or to provide additional information about the protocol specific information associated with the call. This information consists of:
 - The type of call control information elements (ISDN, for example).
 - A character string that contains the protocol specific information elements. For example, in ISDN, the information may include Bearer Capability, Subaddress (for both calling and called devices), High Layer compatibility, and Low Layer compatibility as defined in IS 11572: 1993.

Refer to 12.2.16, “MediaCallCharacteristics”, for a description of the Media Characteristics that are used in Call Control services to select devices for a call and in Call Control events to report the media Characteristics associated with the devices involved with the call.

3. **CSTA Device Identifier** - Each device that can be observed and/or manipulated shall be referenced across the CSTA Service Boundary. To accomplish this, each device shall be identified using a Device Identifier.
 - Throughout this standard, the term *Device Identifier* shall always mean *CSTA Device Identifier*.
 - Device Identifiers may be static or, only when used in the context of a connection identifier, dynamically-assigned.
 - A static Device Identifier shall be stable over time. It shall remain constant and unique between calls, associations and within both the switching and computing functions. An example of a static Device Identifier is an ITU-T E.164 Directory Number.
 - It may be useful for the Switching Function to convert a Device Identifier to another static form for use in service interactions. An example, it might be useful to transform a Public Directory Number into 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 shortened form of Device Identifier is known as a CSTA Short Form Device Identifier.
 - A static Device Identifier may be used in conjunction with “MediaCallCharacteristics”, as specified in 12.2.16, “MediaCallCharacteristics”, on page 105, in order to distinguish among CSTA Devices that share a Device Identifier.
 - A dynamically-assigned Device Identifier is temporary (lasting for the duration of a call) and may be created at any appropriate time. Once a CSTA Device has been included in a call, it may be desirable to continue to refer to the particular instance of the CSTA Device associated with this call for manipulation or tracking. A static Device Identifier may not always be sufficient because it may not be available or because it is too long and cumbersome for efficient use. In these cases the Switching Function can dynamically assign a Device Identifier as a device reference or handle for the duration of the call. Management of the dynamically-assigned Device Identifier is discussed in 6.1.8.
 - The **Device Identifier Status** indicates if an actual Device Identifier is being provided in a parameter or the reason why it is not being provided. The set of possible values for the Device Identifier Status is:
 - *Provided* - A Device Identifier is present.

- *Not Known* - Indicates that the switching function cannot provide the Device Identifier but knows that the device exists.
 - *Not Required* - Indicates that the device is not relevant in this case.
 - *Not Specified* - Indicates that the device cannot be specified.
- The parameter type associated with a particular Device Identifier determines how it is interpreted, restrictions on its use, and the Device Identifier Statuses that are applicable. These parameter types (AssociatedCalledDeviceID, AssociatedCallingDeviceID, CallingDeviceID, CalledDeviceID, DeviceID, RedirectionDeviceID, and SubjectDeviceID) are specified in 12.3, “Identifier Parameter Types”, on page 111.
 - The format of a Device Identifier is specified in Clause 10, “CSTA Device Identifier Formats”.
4. **Device State** - A CSTA Device itself does not have a state or status directly associated with it. The elements, components and calls associated with the CSTA Device do have states and statuses associated with them. The following is the list of these states and statuses associated with a CSTA Device:
- A connection state is the state of a CSTA Device’s logical element’s connection into a call. This state is associated with Call Control features/services. For more information on connection states, refer to Clause 6.1.5, “Connection”, beginning on page 33.
 - The status of the physical components associated with a physical element of a CSTA Device. (e.g., the hookswitch status). For more information, refer to Clause 21, “Physical Device Features”, beginning on page 390.
 - The status of the logical device features/services associated with a logical element of a CSTA Device. (e.g., the forwarding and do not disturb status). For more information, refer to Clause 22, “Logical Device Features”, beginning on page 439.
5. **Device Elements** - A CSTA Device represents various types of telephony endpoints in a switching sub-domain and allows access to telephony services. A CSTA Device can range from a single endpoint (e.g., station) to a set of associated endpoints that form a group. Each CSTA Device is represented by its attributes (e.g., identifier, state(s), type) as well as its features/services. These attributes/features/services are grouped into two categories which are referred to as *device elements*. A device element encompasses the control and observation of a specific set of CSTA Device attributes/features/services. The device elements are: *physical element* and *logical element*.

The logical element of a CSTA Device encompasses the set of attributes/features/services (e.g., Make Call, Set Forward) that have any association with the control and observation of a call at a CSTA Device (i.e., connection). The physical element of a CSTA Device encompasses the set of attributes/feature/services that have any association with physical components of the CSTA Device that would potentially make up the user interface of the device.

All addressable (i.e., has a single identifier) CSTA Devices consist of one of the following device element combinations.

Figure 6-2 Logical Element Only

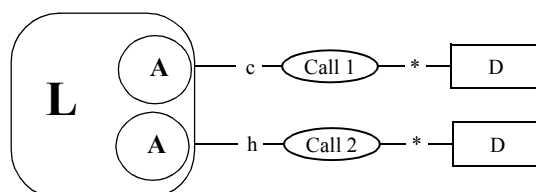


Figure 6-3 Physical Element Only

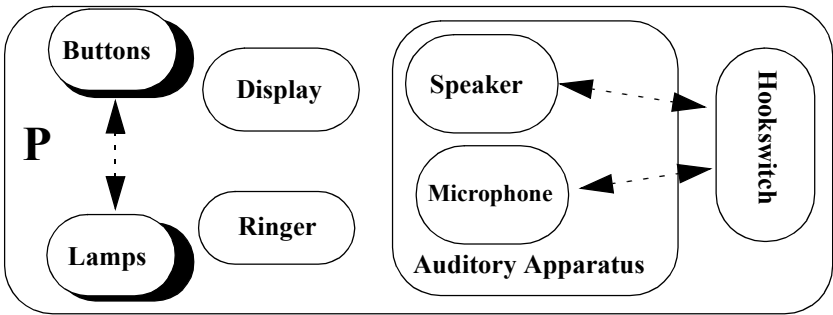
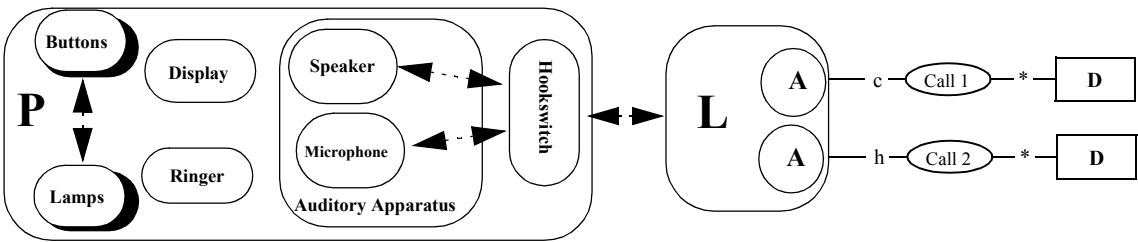
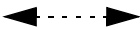


Figure 6-4 A Logical and Physical Element



- D** represents another device
- P** represents the identifiers for the physical elements
- L** represents the identifiers for the logical elements
- A** represents an appearance of a logical element



indicates that there is an interaction and/or association between the elements or components of an element.

For example, a “plain old telephone set” (POTS) consists of a logical and physical element. A computing function learns about the devices, their elements, and their associated attributes/features/services in a switching sub-domain by using the capabilities exchange services (refer to 13.1 beginning on page 122). The following sections describes the device elements and the device attributes/features/services that are associated with the elements in detail.

6.1.3.1 Physical Element

The physical element represents the attributes of the physical components and their associated features/services that make up the user interface of a device (e.g., the components of a telephone set). A physical component at a CSTA Device can be a piece of hardware or a virtual (e.g., software) representation of a piece of hardware. For example, a set of buttons on a device (i.e., physical components of the device) could be comprised of a piece of hardware with 12 buttons and a switching function software representation of 12 more buttons. As a result, the device has a set of 24 buttons associated with its physical element.

The combinations of physical components associated with the physical element are switching function specific. The following features/services are controlled and observed through the physical element:

- The Physical Device features/services, such as Button Press, Get/Set Hookswitch Status, and Get/Set Speaker Volume.
- The I/O Services such as Start Data Path and Send Data (when applied to the physical element of a device).

Note that these features/services also include the associated events and monitoring of these events.

The physical element of a device is observed and/or controlled within the switching function through an assigned Device Identifier.

Note that if the device is a combination of logical and physical elements, the assigned Device Identifier is the same for both elements.

In order for the physical components to interact or be associated with calls at the device, the device shall have a logical device element (for more details, see 6.1.3.2, “Logical Element”) and/or some association(s) with a logical element(s) from another device(s) (for more details, see 6.1.3.3, “Device Configurations”). The physical components interact with calls through these logical device element(s) but it is device and switching function specific as to how these components actually interact and are associated with the calls.

The following sections describe the physical components that can be associated with the physical element of the device. All physical components shall be controlled and observed in conjunction with the physical element (i.e., associated with the physical element’s Device Identifier).

6.1.3.1.1 Auditory Apparatus

An auditory apparatus is a component which is used to convert electronic signals into voice/speech (i.e., a speaker) and/or convert voice/speech into electronic signals (i.e., a microphone) but at a minimum shall have either a speaker or a microphone. A physical element can have several auditory apparatuses associated with it. Each auditory apparatus can be used independent of each other. An auditory apparatus has several attributes which can be controlled and observed by a computing function. The following are those attributes:

1. *Auditory Apparatus type* - There are several types of auditory apparatuses. Each type representing a different physical configuration and/or function. The following is the list of auditory apparatus types:
 - a. *Handset* - An auditory apparatus that is held in a person’s hand and contains a microphone and speaker.
 - b. *Headset* - An auditory apparatus that is worn on a person’s head and contains a microphone and speaker.
 - c. *Speakerphone* - An auditory apparatus that does not require a person’s body to come into contact with the apparatus and contains a microphone and speaker.
 - d. *Speaker-only phone* - A Speakerphone without a microphone.
 - e. *Microphone-only* - A type that provides only a microphone.
 - f. *Other* - An auditory apparatus that is unique to the given switching function.
2. *Auditory Apparatus Identifier* - Each auditory apparatus that can be observed and/or controlled within the switching function is referenced using an assigned identifier. The auditory apparatus identifiers associated with a given physical element’s Device Identifier are unique. This identifier is used to control and observe all auditory apparatus attributes except the hookswitch which is associated with the apparatus.
3. *Microphone* - The auditory apparatus may or may not have a microphone. If a microphone is present at the auditory apparatus, then there are two features of the microphone that may or may not be controlled (i.e., settable) and observed (i.e., readable).
 - a. *Gain* - This is the level at which the microphone is generating the electronic signal. For more details, refer to the definition of the microphoneGainValue parameter in associated services and events.
 - b. *Mute* - This is the capability to temporarily disable the microphone. For more details, refer to the definition of the microphoneMute parameter in associated services and events.

Both of these features are controlled and observed using the auditory apparatus identifier.
4. *Speaker* - The auditory apparatus may or may not have a speaker. If a speaker is present at the auditory apparatus, then there are two features of the speaker that may or may not be controlled (i.e., settable) and observed (i.e., readable).
 - a. *Volume* - This is the level at which the speaker is boosting the electronic signal when generating the associated voice sound waves. For more details, refer to the definition of the speakerVolumeValue parameter in associated services and events.

- b. *Mute* - This is the capability to temporarily disable the speaker. For more details, refer to the definition of the *speakerMute* parameter in associated services and events.

Both of these features are control and observed using the auditory apparatus identifier.

- 5. *Hookswitch association* - This identifies the particular hookswitch that is used to activate (i.e., put into use) and deactivate (i.e., remove from use) the auditory apparatus. It also indicates, if the particular hookswitch can be controlled (i.e., settable) and observed. For more details on hookswitches, refer to associated services and events.

6.1.3.1.2 Hookswitch

A hookswitch is a component which is used to activate (i.e., put into use or off-hook) or deactivate (i.e., remove from use or on-hook) an auditory apparatus(es). When a hookswitch is off-hook, it enables the auditory apparatus(es) to transmit and receive the electronic signals associated with sound, and when it is on-hook, this capability is disabled. A physical element can have several hookswitches associated with it. Each hookswitch can be used independently of each other. A hookswitch has one attribute which can be controlled and observed by a computing function. This attribute is the hookswitch identifier. This identifier is assigned by the switching function. The hookswitch identifiers associated with a given physical element's Device Identifier are unique. This identifier is used to control and observe the status of the particular hookswitch (i.e., on-hook, off-hook).

6.1.3.1.3 Button

A button is a component which executes a specific feature/service that is assigned to it. The most common implementation of this component, is a piece of hardware that is pressed and released, thereby executing the feature/service assigned to it (e.g., each of the number buttons on a station). However, an implementation can use any component that can produce a similar behaviour. A button can also have the capability of toggling between two settings of a feature/service (e.g., enabling and disabling Do Not Disturb, that is you press the button once, it enables the Do Not Disturb feature/service, and press again, it disables the feature/service). The button can also have the capability of looping through a series of features/services. Almost any feature/service or set of features/services can be assigned to a button, but generally a switching function makes visible buttons only with features/services that are not available through the components/attributes/features/services defined in this International Standard. A button can also be used to represent another physical component (e.g., a hookswitch). A physical element can have many buttons associated with it. Each button can be used independently of each other. A button has the following attributes which can be controlled and observed by the computing function:

1. *Button Identifier* - Each button that can be observed and/or controlled within the switching function is referenced using an assigned identifier. The button identifiers associated with a given physical element's Device Identifier are unique. This identifier is used to control and observe all other button attributes.
2. *Button Label* - This is a label by which people interacting with the physical device refer to a given button. This label is a character string which is retained by the switching function. This attribute can also be changed (if supported) by the computing function. The meaning of a Button Label is specific to the users of a particular device and changing it does not change the function of the button.
3. *Button Function* - This is a feature which can be assigned by the switching function to describe the function associated with a given button. The switching function may reassign the functionality of a button and change this attribute as required (in response to other button presses, for example) but it may not be changed directly by the computing function.
4. *Button Associated Number* - This is a diallable digits format Device Identifier which is associated with the feature/service assigned to the button. This Device Identifier is used by the feature/service when it is executed by the button being pressed. This attribute is optional and only applies to buttons that have some form of associated number. This Device Identifier is initially assigned by the switching function and can be changed (if supported) at any time by either the switching or computing function.
5. *Button Press Indicator* - This indicates if the button can be pressed via the Button Press service.

6.1.3.1.4 Lamp

A lamp is a component that represents (i.e., indicates), for example, the status of, for example, a feature/service, physical component, logical device element or other CSTA device. The most common implementation of this

component is a piece of hardware that emits light. However, an implementation can use any component that can produce a similar behaviour (e.g., an icon presented on a display). A physical element can have many lamps associated with it. Each lamp can be used independently of each other. A lamp has several attributes which can be controlled and observed by the computing function. The following are those attributes:

1. *Lamp Identifier* - Each lamp that can be observed and/or controlled within the switching function is referenced using an assigned identifier. The lamp identifiers associated with a given physical element's Device Identifier are unique. This identifier is used to control and observe all other lamp attributes.
2. *Lamp Label* - This is a character string which is assigned by the switching function to describe the feature or service's status associated with this lamp. This attribute cannot be changed by the computing function. The meaning of the Lamp Label attribute is switching function specific.
3. *Lamp Mode* - This is the output of the lamp which is used to indicate the status of the (feature/service) or physical component. The output values are represented by the various ways light can be produced from a lamp. This output can be changed at any time by either the switching or computing function. For more details, refer to definition of the lampMode parameter.
4. *Lamp Brightness* - This attribute indicates the visible brightness of the lamp when it is on. This attribute can be changed by the switching function or by the computing function. For more details, refer to the definition of the lampBrightness parameter.
5. *Lamp Color* - This attribute is an additional characteristic of the lamp which helps distinguish it from other lamps. This attribute can only be changed by the switching function. For more details, refer to the definition of the lampColor parameter.
6. *Button Association* - This identifies a button that is associated with the lamp. The lamp can be used to represent either the status of the feature/service associated with the button or to represent the status of the toggle sequence.

6.1.3.1.5 Ringer

A *ringer* is a component associated with the physical element that provides indication that a device is being rung. There may be one or more ringers associated with a device.

A ringer has attributes that can be controlled and observed by a computing function. The following is a list of those attributes:

1. *Ringer Identifier* - Each ringer that can be observed and/or controlled within the switching function is referenced using an assigned identifier. The ringer identifiers associated with a given physical element's Device Identifier are unique. This identifier is used to control and observe all ringer attributes associated with the device.
2. *Ring Mode* - This attribute describes if the ringer is engaged in a ringing cycle. It will remain *ringing* for the entire ringing cycle (e.g. across consecutive instances of a ringing pattern). Typically only one ringer on a physical element can be rung at one time.
3. *Ring Count* - This attribute describes the number of ring cycles (instances of ring pattern) that the ringer has completed. This attribute is set to 0 immediately before the first ring cycle starts. Note that this is used to query the most recent ring count even after ringing has ceased.
4. *Ring Pattern* - This attribute describes the type of Ring Pattern associated with a ringer. Each individual Ring Pattern cycle may consist of zero or more periods of audible ringing followed by a silent phase. The Ring pattern may be used to help audibly distinguish the types of calls at a device or to uniquely identify a ringer. The meaning of the Ring Pattern is switching function specific.
5. *Ring Volume* - This is the level at which the ringer is set to ring. This information is associated with the ringer until it is reset by the switching function or until it is changed via the Set Ringer Status service.

6.1.3.1.6 Display

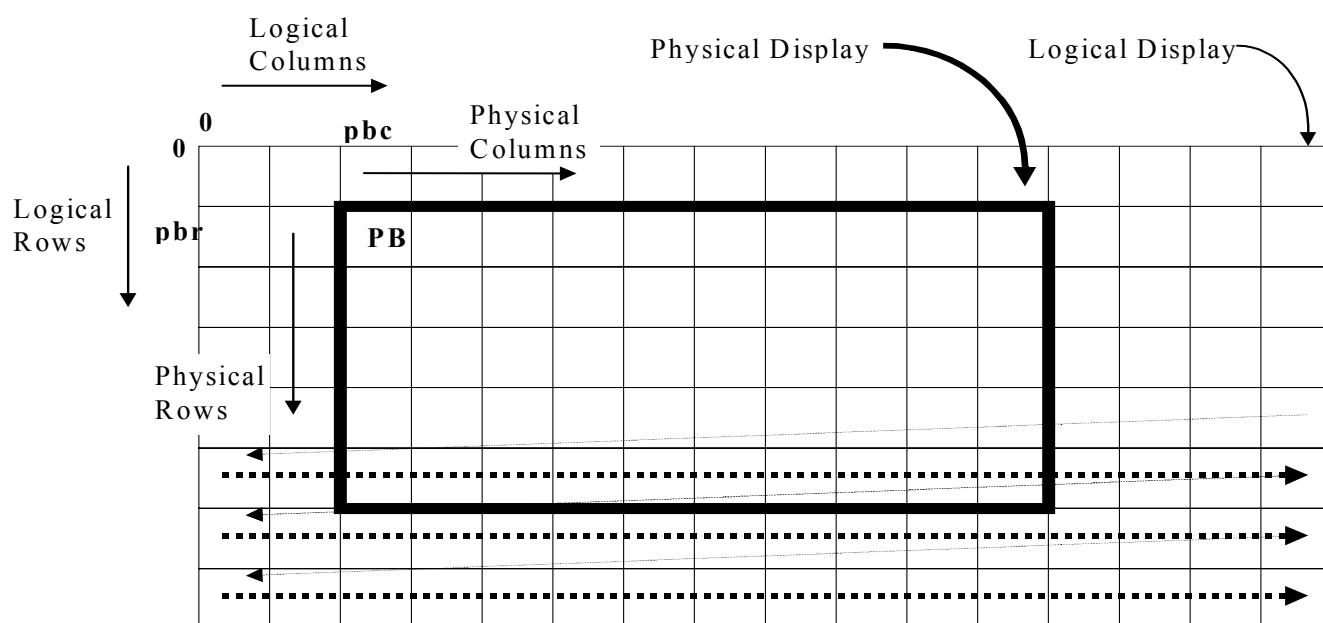
A *display* is a component which presents a two dimensional array of characters associated with the physical element. A physical element can have several displays. A display may be real or virtual; that is it may or may not

actually be present on the physical device itself. Displays have eight attributes as visualized in Figure 6-5, "Display Attributes":

1. *Display Identifier* - To identify a specific display on a physical device.
2. *Logical Rows* - The number of rows on the logical display.
3. *Logical Columns* - The number of columns on the logical display.
4. *Physical Rows* - The number of rows on the physical display. This number is always smaller or equal to *Logical Rows*.
5. *Physical Columns* - The number of columns on the physical display. This number is always smaller or equal to *Logical Columns*.
6. *Physical Base* - The location of the first character of the physical display expressed as (LogicalRowNbr, LogicalColumnNbr) and identified in Figure 6-5 as (**pbr,pbc**). Note that the top-left most position in the logical display is defined as (**0,0**).
7. *Character Set* - Normally ASCII, but may be also be Unicode or a proprietary character set used by the switching function. This attribute is fixed for a given display.
8. *Contents* - A character string which represents the contents of the logical display. Spaces are always present so the size of this string is always the product of the number of logical rows and logical columns. The *contents* can be observed and/or set (if supported) by the computing function. When setting the value of the *Contents*, an offset (starting point in logical display) and a length field (number of characters to be set) shall be given).

The following figure visualizes these attributes further:

Figure 6-5 Display Attributes



.....➔ = Order in which characters from *Contents* are located on the logical display

PB = Physical Base
Located on logical display at location (**pbr,pbc**)

6.1.3.2 Logical Element

A logical element is the part of a device that is used to manage and interact with calls at a device. This element represents the isochronous media stream channels (e.g., ISDN bearer channels) and associated call handling facilities that are used by the device when involved in a call (i.e., via a connection). If a device also has a physical element, the logical element may interact with the physical element's components in order to convey call information (e.g., via lamps) to the user of the device, to provide/manage the media stream data of the call (e.g., via an auditory apparatus) for the user of the device, and to allow the user of the device to manage the calls (e.g., via buttons). The implementation of this interaction is device and switching function specific. The following are the call and call-related features/services that are controlled and observed through the logical element itself:

- Logical Device features/services which are used to indirectly control calls at the device such as Get/Set Forwarding, Get/Set Do Not Disturb and Get/Set Auto Answer.

Note that these features/services also include the associated events and monitoring of these events.

The logical element of a device is observed and/or controlled within the switching function through an assigned Device Identifier.

Note that if the device is a combination of logical and physical elements, the assigned Device Identifier is the same for both elements.

The following sections describe the attributes and components of the logical element. These attributes/component shall be controlled and observed in conjunction with the logical element (i.e., associated with the logical element's Device Identifier).

6.1.3.2.1 Appearance

An appearance is a receptor which is used to connect with at most a single call at the device. A logical element consists of one or more appearances. Appearances are also sometimes called *call appearances*. The number of appearances that a logical element can have is switching function and device specific. Changes in the number of addressable appearances for a logical element are reflected by the capabilities exchange services (13.1 beginning on page 122). Each appearance can be used independently. The following are the call and call-related features/services that are controlled and observed through an appearance for a particular call:

- Call Control features/services such as Make Call, Deflect Call, Answer Call.
- Call Associated features/services such as Associate Data, Generate DTMF, Generate Telephony Tones.
- Routing Services such as Route Request, Route Select and Route End.
- Media Stream Access such as Attach Media Service, and Detach Media Service.
- I/O Services such as Start Data Path and Send Data (when applied to logical elements of a device.)

Note that these features/services also include the associated events and monitoring of these events.

An appearance has several attributes. The following are those attributes:

1. *Addressability* - The addressability of an appearance refers to whether or not the switching function is explicitly representing the appearance to the computing function.
 - a. *Addressable* - An appearance is addressable if it can be explicitly referenced by the computing function at any time with or without the involvement in a call, through a CSTA static device identifier. Refer to Clause 10, "CSTA Device Identifier Formats" for a description of how addressable appearances are referenced.
 - b. *Non-addressable* - An appearance is non-addressable if it can only be referenced, when it is involved with a call, through a CSTA connection identifier. In this case, the logical element dynamically creates and destroys appearances based on the call activity, call capabilities, and features/services of the device. Once the appearance is created (i.e., associated with a call), the corresponding Connection Identifier shall be used to control and observe the appearance. For example, when a call is presented to the device, the logical element creates an appearance to handle the call.

2. *Appearance Type* - The type of appearance is based on its relationship with other devices. The type of appearance determines the functionality and behaviour associated with the logical element of the device. There are two types of appearances: Standard and Bridged Appearances.

Refer to Annex A for a complete description of the types of appearances and their associated behaviour.

6.1.3.3 Device Configurations

A *device configuration* describes the arrangement of the various elements and appearances that can be directly associated with a given device. Multiple device configurations may be formed from the possible combinations of physical elements, logical elements, and different appearance types.

Device configurations are described in terms of a specific device configuration for a particular device:

1. *Device's element combination* - This indicates whether the device has a physical element only, a logical element only or both a logical and physical element.
2. *Other devices using the physical element* - This indicates the list of devices (i.e., their logical elements) that are using the physical element of the base device.
3. *Other devices using the logical element* - This indicates the list of devices (i.e., their physical elements) that are using the logical element of the base device.
4. *The logical element's appearance addressability* - This is an attribute of the appearances of the logical element of the base device (if the logical element is present).
5. *The logical element's appearance type* - This is an attribute of the appearances of the logical element of the base device (if the logical element is present).
6. *The number of appearances associated with the logical element* - This is an attribute of the logical element of the base device (if the logical element is present).

As a set, these attributes describe the device configuration for a specific device. The following sections illustrate typical examples of device configurations that can exist in a switching sub-domain.

Note that in the following examples, where physical and logical elements form part of the same device, the application of a suffix number to the identifying letter identifies that they are parts of the same device (e.g. L1, P1 are a single device; L1, P2 are elements from different devices).

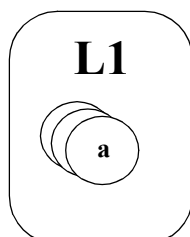
6.1.3.3.1 Logical Element Only

This device configuration has only one logical element (e.g., some Park devices). The following identify the attributes of this device configuration:

- *Device's element combination* - logical element only (L1)
- *Other devices using the physical element* - None
- *Other devices using the logical element* - None
- *The logical element's appearance addressability* - Non-addressable
- *The logical element's appearance type* - Selected-standard
- *The number of appearances associated with the logical element* - unlimited (switching function based limits)

Figure 6-6 is a diagram of a logical element only device configuration.

Figure 6-6 Logical Element Only Device Configuration



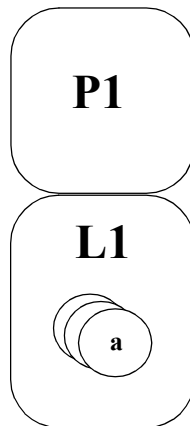
6.1.3.3.2 Single Physical and Logical Element

A *Single Physical and Logical Element* device configuration consists of a single physical element of the device associated with a single logical element of the device that contains non-addressable standard appearances. This device configuration could be used to model a basic telephone station device (e.g., a Plain Old Telephone Service (POTS) telephone or a featured telephone). The following identify the attributes of this device configuration:

- *Device's element combination* - both a logical and physical element (L1/P1)
- *Other devices using the physical element* - None
- *Other devices using the logical element* - None
- *The logical element's appearance addressability* - Non-addressable
- *The logical element's appearance type* - Selected-standard
- *The number of appearances associated with the logical element* - unlimited (switching function based limits)

Figure 6-7 is a diagram of this device configuration.

Figure 6-7 Single Physical and Logical Element Device Configuration (One Device)



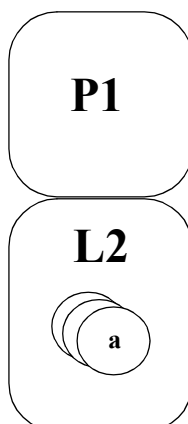
In this figure, the labels “L” and “P” represent the logical and physical elements of the device respectively.

Another variation of the single logical and physical element device configuration involves two different devices - one with a logical element only and one with a physical element only - that are associated with each other. From the perspective of the physical device element P1, the device configuration in this example can be represented as follows:

- *Device's element combination* - physical element only (P1)
- *Other devices using the physical element* - one device (L2)
- *Other devices using the logical element* - None
- *The logical element's appearance addressability* - N/A
- *The logical element's appearance type* - N/A
- *The number of appearances associated with the logical element* - N/A

Figure 6-8 is a diagram of this device configuration.

Figure 6-8 Single Physical and Logical Device Configuration (two devices)



6.1.3.3.3 Multiple Logical Elements

A *multiple logical elements* device configuration consists of a single physical element associated with multiple logical elements containing standard appearances. A multi-line telephone station could be modeled using this device configuration. The following identify the attributes of this device configuration:

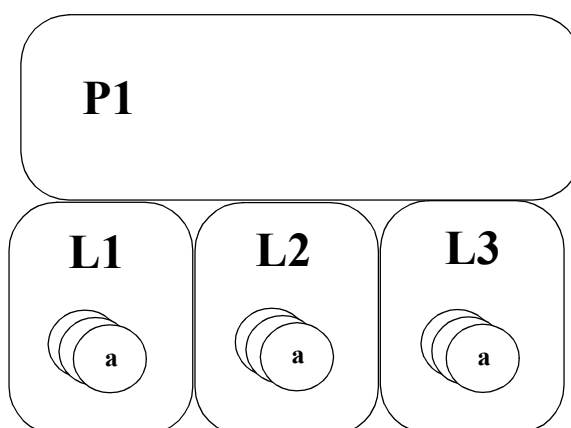
- *Device's element combination* - physical and logical element combination (L1/P1)
- *Other devices using the physical element* - two devices (L2,L3)
- *Other devices using the logical element* - None
- *The logical element's appearance addressability* - non-addressable
- *The logical element's appearance type* - Selected-standard
- *The number of appearances associated with the logical element* - unlimited (switching function based limits)

None of the logical elements in this device configuration need to be part of the same device as the physical element.

Multiple logical elements device configuration represents a single physical element (a telephone set) in a telephone system that supports only one appearance per logical element but has access to multiple calls simultaneously.

Figure 6-9 is a diagram of a multiple logical elements device configuration.

Figure 6-9 Multiple Logical Elements Device Configuration



6.1.3.3.4 Multiple Appearance

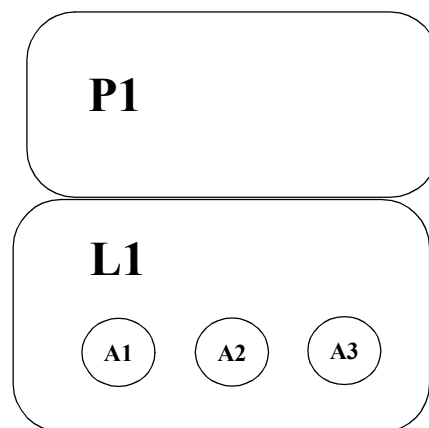
A *multiple appearance* device configuration consists of single physical element and a single logical element containing two or more addressable appearances. Multiple appearance device configurations are another way to represent a single telephone set that has access to multiple calls simultaneously. This approach could be used in a

telephone system that supports addressable standard appearances. This device configuration is sometimes called a *call appearance station*. The following identify the attributes of this device configuration:

- *Device's element combination* - physical and logical element combination (L1/P1)
- *Other devices using the physical element* - None
- *Other devices using the logical element* - None
- *The logical element's appearance addressability* - addressable
- *The logical element's appearance type* - Selected-standard
- *The number of appearances associated with the logical element* - 3 (A1, A2, A3)

Figure 6-10 is a diagram of a Multiple Appearance Device Configuration.

Figure 6-10 Multiple Appearance Device Configuration



6.1.3.3.5 Bridged

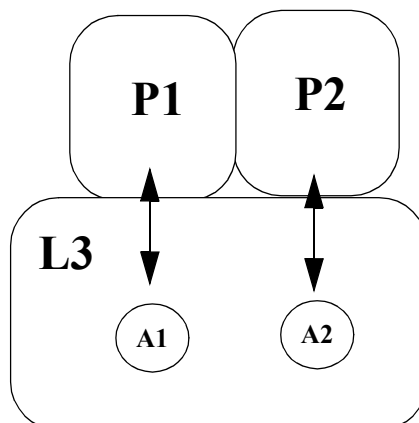
A *bridged* device configuration involves bridged appearances. The characteristics of a bridged device configuration depends upon whether the device configuration is for a physical or logical element.

In the example presented in Figure 6-11, the device configuration shown is for logical element L3 which has bridged appearances. The following identify the attributes of this device configuration:

- *Device's element combination* - logical element only (L3)
- *Other devices using the physical element* - None
- *Other devices using the logical element* - two devices (P1,P2)
- *The logical element's appearance addressability* - addressable
- *The logical element's appearance type* - Independent-shared-bridged
- *The number of appearances associated with the logical element* - 2 (A1 for P1 and A2 for P2)

Figure 6-11 is a diagram of a Bridged Device Configuration.

Figure 6-11 Bridged Device Configuration

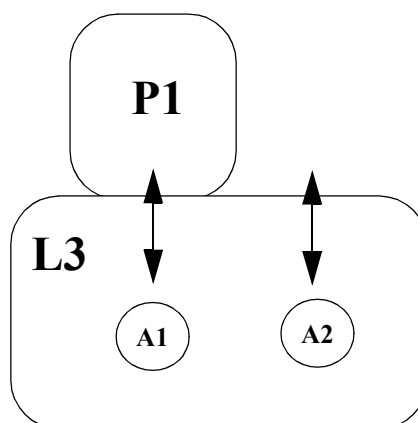


The device configuration for the physical element P1 in this example, is shown in Figure 6-12. In P1's device configuration there are only two device elements rather than three; one is the physical element (P1) and one is the logical element which has two addressable bridged appearances. The following identify the attributes of this device configuration:

- *Device's element combination* - physical element only (P1)
- *Other devices using the physical element* - one device (L3 using appearance A1)
- *Other devices using the logical element* - None
- *The logical element's appearance addressability* - N/A
- *The logical element's appearance type* - N/A
- *The number of appearances associated with the logical element* - N/A

Figure 6-12 is a diagram of a Bridged Device Configuration.

Figure 6-12 Bridged Device Configuration



6.1.3.3.6 Hybrid

A physical element associated with multiple logical elements that each have different types of appearances has a *hybrid* device configuration.

An arbitrary example of a hybrid device configuration is shown in Figure 6-13. This example consists of one physical element and three logical elements.

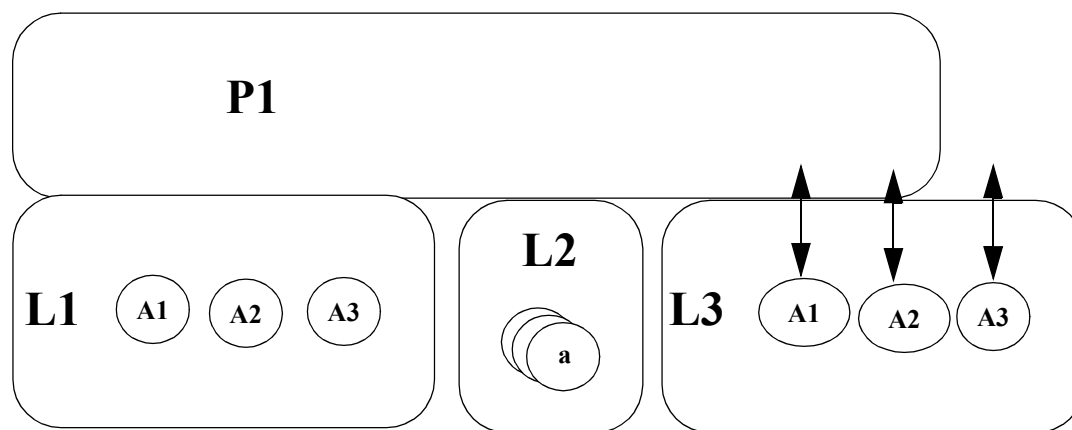
- device 1 has both a physical element (P1) and a logical element (L1) containing two addressable standard appearances
- device 2 has only a logical element (L2) with non-addressable standard appearances
- device 3 has only a logical element (L3) with three addressable bridged appearances

The following identify the attributes of this device configuration:

- *Device's element combination* - logical and physical element (L1/P1)
- *Other devices using the physical element* - two devices (L2, L3 using appearances A1 and A2)
- *Other devices using the logical element* - None
- *The logical element's appearance addressability* - addressable
- *The logical element's appearance type* - Selected-standard
- *The number of appearances associated with the logical element* - 3 (A1, A2, A3)

Figure 6-13 is a diagram of a Hybrid Device Configuration.

Figure 6-13 Hybrid Device Configuration



6.1.3.4 Device Categories

The device category of a particular device provides a generic indication of the device's behaviour and configuration. The computing function should use the device category along with other information provided by the capabilities exchange services to model a given device.

6.1.3.4.1 Station Device Category

This category of device can range from a basic "Plain Old Telephone Set" (POTS) device to a very complex feature telephone device. Station devices can be represented by any single device configuration type, or more commonly, as a hybrid of two or more different device configuration types. The physical component, if present, may have any combination of components. The logical element(s) may have any number of appearances appropriate for the type of device configuration.

6.1.3.4.2 Network Interface Device Category

A *Network Interface Device* is a category of device which is within the switching sub-domain and is connected to another telephone network.

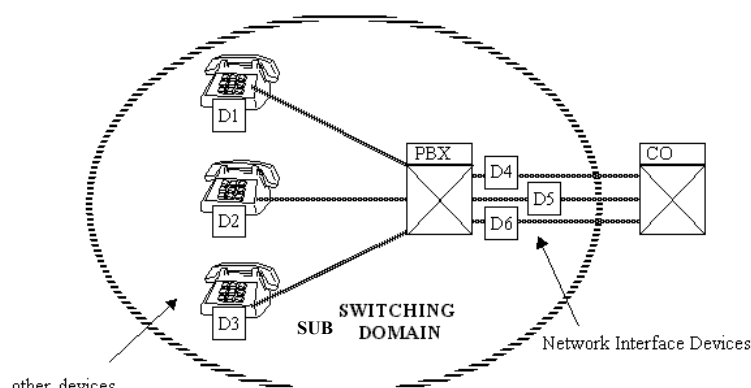
A given switching sub-domain is connected to another telephone network(s) (which may or may not be thought of as other switching sub-domains) through one or more Network Interface Devices.

To indicate when a given call involves a Network Interface Device and an external device, the switching function provides a Network Reached event to the computing function specifying what Network Interface device is being used (if known), and what subsequent call-related information is subject to the capabilities of the network being used.

The following are examples which illustrate the use of Network Interface Devices.

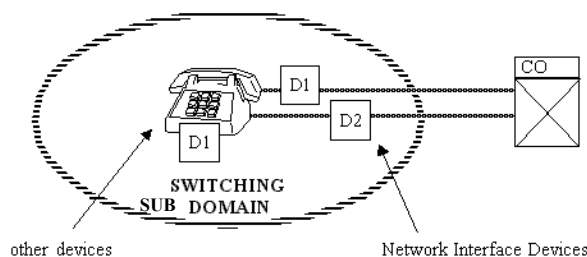
In Figure 6-14, the switching sub-domain is centered on a PBX. The Network Interface Devices are distinct devices and are commonly referred to as "trunks."

Figure 6-14 Trunks As Network Interface Devices



In Figure 6-15, the switching sub-domain is an individual telephone station connected directly to the public telephone network. In this example, the two Network Interface Devices represent two Central Office lines.

Figure 6-15 Central Office Lines As Network Interface Devices



6.1.3.4.3 ACD Device Category

An ACD (Automatic Call Distributor) is a device that distributes calls. An ACD device only consists of the distribution mechanism and may be associated with the devices to which the mechanism distributes calls.

A dynamic process, *Agent Log On* allows an association between an ACD device and a distributed-to device to be created, removed and changed at anytime. Refer to 6.1.3.6, “Agent”, on page 24 for a description of Agent and the Agent Log On Process.

An ACD device is represented as a logical element only device configuration. When a call is presented to an ACD device, a connection in the entering distribution mode of the alerting state is created. Calls that are presented to the ACD device may be queued before they are distributed. The conditions under which the call is queued are switching function specific. Many calls can be simultaneously enqueued, pending their distribution. The ACD device can, for example:

- distribute a call to an agent that had logged on to the ACD device
- distribute a call to an agent that had logged on to an ACD Group
- distribute a call to another ACD device (as well as any other type of internal or external device)
- queue a call to an ACD Group, or to resources that provide message playing, message prompting, voice response interaction, etc.

There are two ACD device models - visible ACD-related devices and non-visible ACD-related devices (note that an ACD device can support one or both models):

- *Visible ACD-Related Devices* - In this model, the ACD device, and the devices (e.g., voice announcement units, devices the call may queue to, other ACD devices, ACD groups, etc.) that can interact with the ACD device while the call is being handled by the ACD device, can be monitored/ controlled and are represented uniquely by the switching function with their own connection when associated with a call.

- *Non-Visible ACD-Related Devices* - In this model, the ACD device, and the devices (e.g., voice announcement units, devices the call may queue to, other ACD devices, ACD groups) that can interact with the ACD device while the call is being handled by the ACD device, are represented by the switching function using a single connection at the ACD device. Neither these devices, which interact with the ACD device, nor their connections to calls queued at the ACD device can be monitored or controlled. Note that when a single connection is used in this manner, all relationships between the call and the ACD device are terminated when the call is no longer part of the ACD device (diverted from the device, for example).

6.1.3.4.4 Park Device Category

A *park* device category is a device that is exclusively used by the switching function to park calls on behalf of other devices in the switching sub-domain. These calls, once parked, may be retrieved by a device in the switching sub-domain. The number of calls that can be parked at one of these devices is switching function specific. These devices can be represented by either a single physical and logical elements device configuration or a logical element only device configuration. The visibility of these devices within the switching sub-domain is switching function specific.

Note that calls may be parked at devices other than a Park device.

6.1.3.4.5 Group Device Category

A *group* device models a relationship between CSTA devices that is characterized by the fact that these devices share a common device identifier. This relationship can be permanent or temporary. The group model differs from the bridged appearances model in that the relationship is between devices' logical elements while for bridged appearances the relationship is between a logical element and a physical element.

The group device has a logical element that has relationships with the logical elements of the devices that are members of this group. The group device is referenced by the group device identifier.

A group device may have a distribution mechanism. The distribution mechanism may support queuing. When a group device has a distribution mechanism, the group device identifier represents both the distribution mechanism and the member devices.

Group Device Attributes

The attributes of a group device are (a group device may have more than one of these attributes (e.g., a hunt and pick group device):

- **Hunt** - The hunt attribute characterizes a group device (also called hunt group) that has the capability to distribute calls to the member devices according to different selection modes (e.g., cyclical, sequential, longest idle time). The association between the group device and the member devices is always fixed by the switching function. A hunt group may have the capability to queue these calls before they are distributed.
- **Pick** - The pick attribute characterizes a group device (also called pick group) that represents a collection of devices that can be used with the Group Pickup Call service. When a call is delivered to a device in the pick group, other devices in the group can be used to answer the particular call (i.e., picking the call). When the call is picked, it is diverted from the originally delivered device and connected to the device which picked the call. The details associated with the pick group feature are documented in the Group Pickup Call service. Only devices associated with the pick group device can pick a call from the given group unless the pick group device is addressable within the switching sub-domain and the pick group device is configured to allow devices outside the group to pick calls. The addressability of the pick device within the switching sub-domain is switching function specific. If the pick group device is addressable, its Device Identifier can only be used in conjunction with the Group Pickup Call service (i.e. the computing function has limited control and can not observe the pick group device). The number of devices that are associated with the pick group device is switching function and device specific. The association between the pick group device and the other devices is not visible within the switching sub-domain (i.e., does not have an explicit representation). The association between the pick group device and the other devices is fixed by the switching function. Changes in the number of devices associated with the pick group device are reflected by the capabilities exchange services.

- ACD - The ACD attribute characterizes a group device (also called an ACD group) that has a distribution mechanism similar to a hunt group. In addition, this group device represents an explicit association between the distribution mechanism (the ACD) and the distributed-to-devices. A dynamic process, *Agent Log On* allows an association between an ACD group and the distributed-to devices to be created, removed and changed at anytime. Refer to 6.1.3.6, “Agent”, on page 24 for a description of Agent and the Agent Log On Process.
- Other - The Other attribute characterizes a group device whose characteristics are switching function-dependent.

Group Device and Monitoring

Monitoring of a group device in which the group device includes a distribution mechanism may follow either the *group inclusive* or the *group exclusive* model. The switching function may support either model, or both models simultaneously (with individual group devices supporting either the inclusive or exclusive model). The model(s) supported by a switching function is indicated by the Get Switching Function Capabilities service.

In the group inclusive model, the scope of a monitor on the group device includes the distribution mechanism and all member devices. In the group exclusive model, the scope of the monitor on the group device includes only the distribution mechanism.

The events reported to the monitor of a group device are the same as those reported to monitors of the individual member devices, with the following exceptions:

- A single Monitor Start service request is set on the group device identifier.
- The events for the member devices, if they are in the scope of the monitor, are reported with the cross reference identifier of the monitor set on the group device.
- When an incoming call is received, the called device represents the group device and the subject device (e.g., alerting device) represents the member device.

In addition, if the group device supports a distribution mechanism, all events reported for the call when it is present at the distribution mechanism (e.g., queued), are reported with a connection identifier consisting of the call identifier and the group device identifier. When the call leaves the distribution mechanism to be delivered to a member device, a Diverted event is generated.

Some services applied to the group device are applied to all member devices of the group device (e.g., Set Agent State). This capability is indicated by the capabilities exchange services.

6.1.3.4.6 Other Device Category

A device in this category has characteristics (i.e., device configuration, capabilities, type of elements, element components) which are switching function specific.

6.1.3.5 Named Device Types

Switching Function implementations may indicate that a device is of a particular device type. The following device types may be used for this purpose but the interpretation of a given device type is implementation specific.

- ACD
- ACD Group
- Button
- Button Group
- Conference Bridge
- Line
- Line Group
- Operator
- Operator Group

- Parking Device
- Station
- Station Group
- Trunk
- Trunk Group
- Other
- Other Group

6.1.3.6 Agent

An agent represents a device's association and activities with one or more ACD devices or ACD groups.

An agent becomes associated with a specific ACD device or ACD group by the process of logging on. There are two Agent Log On Models that may be supported by a switching function (as indicated by the capability exchange services). Note that multiple models may be supported by a switching function and by a single agent.

- *Log On to an ACD device* - In this model an agent logs on to ACD device and becomes associated with the activities of the ACD device. There is no association with any ACD group. This may be achieved by using the Set Agent State (loggedOn) without providing the ACD group parameter. Note that this capability cannot be simultaneously supported with the Log On to an ACD Group (implicit/one step) described below.
- *Log On to an ACD Group* - In this model an agent becomes associated with the activities of an ACD group. This may be accomplished by the following:
 - explicit/one step - by using the Set Agent State (loggedOn) service and supplying the ACD group parameter
 - explicit/two step - by first logging on to an ACD device using the Set Agent State (loggedOn) service and omitting the ACD group parameter (described above). Once logged on to an ACD device, a log on to a specific group is achieved by using another Set Agent State (loggedOn) service with the ACD group parameter provided.
 - implicit/one step - by logging on to an ACD Group using the Set Agent State (loggedOn) service and omitting the ACD group parameter. If supported by the switching function, the agent is automatically logged on to an ACD group. Note that this capability cannot be simultaneously supported with the Log On to an ACD device.

An agent has several attributes that can be controlled and observed by the computing function, as described below.

6.1.3.6.1 Agent Identifier

Each agent that can be observed and/or controlled within the switching function is referenced using an assigned identifier. This identifier is associated with the logical element's Device Identifier of the device. Certain switching functions may not assign an agent identifier to the agent. In these cases, the logical element's Device Identifier is used to represent the agent associated with the device. The format of the agent identifier is switching function specific.

There are two ways that an agent identifier may be provided:

- as a parameter in an event (via the agentID parameter in the Agent Logged Off event, for example) or
- as a sub-string in the Switching Function Representation format of a logical element's Device Identifier, thereby making a unique identifier for the logical element and the associated agent.

When an agent is associated with more than one ACD group and the switching function assigns a different agent identifier for each ACD group that the agent is associated with, then the following applies:

- The agent identifier shall be supplied (via the agentID parameter) on the agent state events associated with this agent (e.g., Agent Ready, Agent Not Ready).

- This agent identifier may or may not be part of the Device Identifier parameters (i.e., the agent identifier sub-string in the Switching Function Representation format) in the call control events that are associated with the agent in an ACD call. Non-ACD calls are unaffected.
- This agent identifier shall be part of the Device Identifier parameter (i.e., the agent identifier sub-string in the Switching Function Representation format) on services when the computing function want to focus the service at a particular ACD call or group, otherwise if not supplied, the switching function will choose which ACD call or group the service is focused.

When an agent is associated with one ACD group, the switching function may or may not assign an agent identifier to the agent. If the agent identifier is not assigned, the logical element's Device Identifier of the agent device is used to represent the agent in the ACD group. If the agent identifier is assigned, then either the logical element's Device Identifier of the agent device or the agent identifier may be used to represent the agent in the ACD group. This also applies to the case where the switching function has only one agent identifier (either assigned or not) for an agent that is associated with multiple ACD groups. When only one agent identifier is assigned to an agent the following applies:

- The agent identifier may be supplied (via the agentID parameter) on the agent state events associated with this agent (e.g., Agent Ready, Agent Not Ready).
- This agent identifier may be part of the Device Identifier parameters (i.e., the agent identifier sub-string in the Switching Function Representation format) in call control events that are associated with the agent in an ACD call. Non-ACD calls are unaffected.
- This agent identifier may be supplied as part of the Device Identifier parameter (i.e., the agent identifier sub-string in the Switching Function Representation format) on services.

Note that when the agent identifier is supplied as part of the Device Identifier and the agent identifier is not associated with the device, the service request shall be rejected with a negative acknowledgement.

6.1.3.6.2 Agent Password

The agent password authenticates the agent's association with a given device and ACD device or ACD group in the switching sub-domain. The agent password is used when the agent is associated with the device and made available to the ACD device or ACD group (i.e., log on) and when the agent is made unavailable to the ACD device or ACD group (i.e., log off). An agent password may be assigned to each agent. For more details, refer to the definition of the agentPassword parameter in associated services.

6.1.3.6.3 Agent Group Association

This identifies the ACD group that is associated with the agent. There can be zero or more ACD groups associated with the same agent. This is represented via the ACD group's logical element's Device Identifier.

6.1.3.6.4 Agent State

A state that an agent may take in relation to an ACD device or ACD group and the calls associated with the ACD device or ACD group.

The computing function is informed of agent state changes through agent state event reporting. These event reports are sent to the computing function when a monitor is started on either the ACD device or ACD group (if supported by the switching function) or the device associated with the agent or agents. The following are the agent states with respect to an agent and a particular ACD device or ACD group:

- *Agent Null* - The state where an agent is not logged-on to (i.e., logged off from) the ACD device or ACD group at a particular device. Logging-on and logging-off from an ACD device or ACD group shall cause the transitions from and to this state. The event report that represents the entry to this state is the Agent Logged Off event. Note that the presence/absence of the ACD Group parameter in this event determines if the agent has logged off of an ACD device or an ACD group
- *Agent Logged On* - The state where an agent is logged-on at a particular device to an ACD device or ACD group and is ready to contribute to the activities of the ACD device or ACD group. This state does not indicate that the agent is ready to accept ACD calls (refer to the Agent Ready event section). The event report that represents the entry to this state is the Agent Logged On event. Note that the presence/absence of the ACD Group parameter in this event determines if the agent has logged on to an ACD device or an ACD group.

- *Agent Not Ready* - The state where an agent is logged-on at a particular device to an ACD device or ACD group but is not prepared to handle calls that the ACD distributes. While in this state an agent may receive calls that are not ACD calls. The event report that represents the entry to this state is the Agent Not Ready event.
- *Agent Ready* - The state where an agent is logged-on at a particular device to an ACD device or ACD group and is prepared to handle ACD calls even though it may be involved with non-ACD calls. The event report that represents the entry to this state is the Agent Ready event.
- *Agent Busy* - The state where an agent is involved with an existing ACD call at the device, even if that call is on hold at that device. Calls between agents, calls between supervisors and agents and private calls may or may not cause this transition. The event report that represents the entry to this state is the Agent Busy event.
- *Agent Working After Call* - The state where an agent is no longer connected to an ACD call but is still occupied with work related to a previous ACD call. In this state, an agent cannot receive ACD calls but may be able to receive non-ACD calls. The agent may be performing administrative duties (e.g., updating a business order form) for a previous call, or may be involved with a non-ACD call. The event report that represents the entry to this state is the Agent Working After Call event.

Note that this International Standard does not impose or restrict the transition of an agent state to another agent state. Any implementation restrictions shall be reflected by a negative acknowledgement to the Set Agent State service request.

6.1.3.6.5 Agent State Models

It is possible that an agent may have several agent states with respect to the different associated ACD devices or ACD groups. Alternatively, an agent may have a single state to describe its relationship to all associated ACD devices or ACD groups (if the state is, in fact, the same for all of them).

The following Multi-State Agent models that indicate how the switching function maintains the agent states for each ACD or ACD group that is associated with an agent. (The capability exchange services indicate which models are supported by a switching function.) The models are:

- *Agent Multi-State Model (Independent Group Working)* - In this model, as illustrated in Table 6-1, the switching function maintains an independent agent state model for each ACD device or ACD group associated with an agent. All agent state transitions are totally independent. The cause code of Forced does not appear since no state is forced. (Note that the numbers between parenthesis in the following figures represent the ACD group number).

Table 6-1 Agent Multi-State Model (Independent Group Working) Illustration

	Step 1	Step 2	Step 3	Step 4	Step 5
ACD Group 1 State	Not Ready (1)	Not Ready (1)	Not Ready (1)	Not Ready (1)	Ready (1)
ACD Group 2 State	Not Ready (2)	Ready (2)	Busy (2)	Working After Call (2)	Ready (2)
ACD Group 3 State	Not Ready (3)	Ready (3)	Ready (3)	Ready (3)	Busy (3)
ACD Device State	Ready	Busy	Busy	Busy	Ready

- *Agent Multi-State Model (Semi-Independent Linked)* - In this model, as illustrated in Table 6-2, the switching function maintains an agent state model for each ACD device or ACD group associated with an agent. However the agent state models for each ACD device or ACD group are linked together. For example, if an agent is associated with two ACD groups (both states are Ready), and the agent connects to an ACD call for the first ACD group, the agent state for the first group transitions to Busy while the agent state for the second

group goes to Not Ready (with a cause code of Forced to indicate that the state transition was the result of another ACD device or ACD group activity).

Table 6-2 Agent Multi-State Model (Semi-Independent Linked) Illustration

	Step 1	Step 2	Step 3	Step 4	Step 5
ACD Group 1 State	Not Ready (1)	Not Ready (1)	Not Ready (1) - Forced cause	Not Ready (1) - Forced cause	Not Ready (1)
ACD Group 2 State	Not Ready (2)	Ready (2)	Busy (2)	Working After Call (2)	Ready (2)
ACD Group 3 State	Not Ready (3)	Ready (3)	Not Ready (3) - Forced cause	Not Ready (3) - Forced cause	Ready (3)
ACD Device State	Ready	Ready	Not Ready - Forced cause	Not Ready - Forced cause	Ready

- *Agent Orientated Model* - In this model, as illustrated in Table 6-3, the switching function maintains one state for the agent, no matter how many ACD devices or ACD groups the agent is associated with. When the ACD Group parameter is included in an agent event, it indicates the ACD group that has caused the agent state transition. When the ACD Group parameter is not included in an agent event, it indicates that non-ACD group activity has occurred.

Table 6-3 Agent Orientated Model Illustration

	Step 1	Step 2	Step 3	Step 4	Step 5
ACD Group 1 State	Not Ready	Ready	Busy	Working After Call	Ready
ACD Group 2 State	Not Ready	Ready	Busy	Working After Call	Ready
ACD Group 3 State	Not Ready	Ready	Busy	Working After Call	Ready
ACD Device State	Not Ready	Ready	Busy	Working After Call	Ready
Single Event	Not Ready (2)	Ready (2)	Busy(2)	Working After Call (2)	Ready (2)

6.1.4 Call

Calls are communication relationships between one or more CSTA Devices. A call's behaviour can be observed and manipulated across the CSTA service boundary (also called service boundary in this International Standard). During some call phases (e.g., establishment and release) the call is not completely formed and there may be only a single CSTA Device involved (for example, the CSTA Device on whose behalf the call was initiated). In some call control operations, such as a conference and transfer, one CSTA Device in a call is replaced with another CSTA Device, or two calls are merged into a single call.

The CSTA Call attributes, which are described in detail in the following sub-clauses, shall be:

- Call Identifier
- Correlator Data
- User Data
- Media Call Characteristics
- Account Information
- Authorisation Code
- Charging Information

6.1.4.1 Call Identifier

A CSTA *Call Identifier* (also called *Call Identifier*) is a reference associated with a call whereby the call is known to the switching, computing and special resource functions through the call's life. A Call Identifier shall be

allocated to each call by the Switching Function, at the latest, when the call first becomes visible across a CSTA Service Boundary. It shall be unique within a switching sub-domain and shall be the same for all CSTA Devices in the call. A Call Identifier can be assigned to a call before the call is fully established. For example, an incoming call may be assigned a Call Identifier when the called CSTA Device is alerting and before the call has been answered. A Call Identifier shall not only reference the entire call within the sub-domain but shall also infer a reference to that part of the call that is outside the sub-domain.

A call can pass through various stages involving many different CSTA Devices before it finally terminates. Some of these stages cause a call to change identifiers. Examples of services that cause this are Transfer and Conference. During the operation of these services, or as a result of manual intervention, the Call Identifier may change as a result of two calls being merged by the switching function but the call shall continue as a CSTA object. This merger results in the Call Identifiers for both old calls changing to a new identifier for the resulting calls in which the CSTA Devices are involved. The respective Conferenced or Transferred event specifies the transition from the old Call Identifiers to the new Call Identifiers indicating the old invalid Call Identifiers. Management of Call Identifiers is covered in section 6.1.8, “Management of Dynamically-Assigned Identifiers”.

6.1.4.2 Call State

The term *Call State* means the collective set of Connection states for all the Connections comprising a call. Call state is returned only by the Snapshot Device Service for CSTA Devices that have calls. Connection states are further described in 6.1.5, “Connection”, on page 33. Call states are described in more detail in 6.1.6, “Call State Definitions”, on page 37.

6.1.4.3 Correlator Data

Correlator Data is computing function specific data which has been attached to a call that a computing function is controlling or observing. This information, for example, might be a key to a database entry, an application command sequence, file name, etc. Once Correlator Data is associated with a call, it remains with the call for its entire duration (at least one CSTA Device is actively involved in the call), or until the computing function overwrites the data with new data. In order to remove data, the computing function shall overwrite the existing data with null data. Correlator Data enters the switching sub-domain in two ways:

1. A computing function provides Correlator Data on a service request (e.g. Call Control and Call Associated services).
2. Correlator Data arrives from an external network connection with a call (for example, Correlator Data may be used as a key to pop a screen for the call). This International Standard defines one possible mechanism for delivering Correlator Data through an external network. This mechanism is described in Annex B.

Permitting a computing function to associate its own information with a call allows multiple computing functions to share data on calls which they are all controlling or observing. This feature is useful when calls are moving from one computing function to another in a distributed computer network or from one switching sub-domain to another. For more information on Correlator Data, refer to 12.2.10, “CorrelatorData”, on page 90.

Correlator Data is provided by the Computing Function and associated with the call for its entire duration or until overwritten with new data. This data survives Conference and Transfer and can be provided on various events. An application may remove the Correlator Data by overwriting existing data with null data.

When Correlator Data is associated with a call, call events that indicate that a CSTA Device has become part of a call (such as Delivered, Established and Queued events, for example) shall include the Correlator Data (if this parameter is supported in the event being reported). Subsequent call events also may contain the Correlator Data.

Note that “null Correlator Data” means Correlator Data information with zero length.

When a consultation call is transferred or conferenced and null or non-null Correlator Data is associated with the secondary call, the Correlator Data in the resulting call shall always be the same Correlator Data that was associated with the secondary call (even if only the primary call had non-null Correlator Data). In that case the Correlator Data

(if any) associated with the primary call is discarded. If the secondary call contains no Correlator Data, the Correlator Data associated with the resulting call is that which was associated with the primary call.

Table 6-4 Inheritance rules for Correlator Data in Conference and Transfer

Secondary Call	Primary Call		
	No Correlator Data	Null Correlator Data	Correlator Data 1
No Correlator Data	No Correlator Data	Null Correlator Data	Correlator Data 1
Null Correlator Data	Null Correlator Data	Null Correlator Data	Null Correlator Data
Correlator Data 2	Correlator Data 2	Correlator Data 2	Correlator Data 2

When Correlator Data is associated with a call via the Associate Data service, the Call Information event is provided by the switching function. If the data is changed by any other service, the switching function does not provide the Call Information event.

6.1.4.4 User Data

User Data is call-related computing function-to-computing function information that, unlike Correlator Data, is not associated with a call for the life of the call. The switching function receives User Data in two ways:

- A computing function sends User Data in a service request.
- User Data arrives from an external network connection with a call (for example, User Data may be used as a key to pop a screen for the call).

Both a computing function and the network may send User Data in two ways:

1. *With Call Control Activity* - Call control service requests (and network signalling for call control) permit User Data as an optional parameter. The switching function reflects the delivery of User Data in the first call control event that results from the switching function or network carrying out the call control activity. When the computing function provides User Data in a Call Control service request, the User Data is delivered to other parties if and only if the call control service successfully completes. If the switching function does not generate the call control event that corresponds to the call control activity because the computing function has set an event filter that filters out the relevant event, then the User Data is not propagated to subsequent events and the User Data information will be lost. Refer to the description of the individual Call Control services for more details on the events that will contain User Data for those services.
2. *Independent of Call Control Activity* - The computing function may use the Send User Information service to pass User Data at any time. Some networks provide an independent signalling mechanism for sending User Data. The switching function generates a Call Information event with the userData parameter containing the received User Data to reflect the delivery of the User Data. Independent of call control activity, this event is generated for all computing functions monitoring the call and all computing functions monitoring any CSTA Devices that have a connection to that call.

This International Standard defines one possible mechanism for delivering User Data through an external network. This mechanism is described in Annex B.

When a computing function uses a service to send User Data, the switching function sends that User Data only after the switching function sends a positive acknowledgement to the service request.

User Data is described further in 12.2.23, “UserData”, on page 109.

6.1.4.5 Other Call Related Information

There is additional information associated with calls such as:

- *Account Information* - A computing function or business-specific piece of information that is assigned to a call for accounting purposes. For more information, see 12.2.1, “AccountInfo”, on page 86.
- *Authorisation Code* - A code provided to the switching function that is used to check if a computing function user is authorised to perform a given service. For more information see 12.2.3, “AuthCode”, on page 86.

- *Charging Information* - An amount charged to a device for a call in which the device was involved. For more information, see 12.2.7, “ChargingInfo”, on page 89.

6.1.4.6 Media Call Characteristics

This International Standard covers the control and observation of calls within either a voice or digital data switching sub-domain (e.g., network) or a switching sub-domain that is a combination of both (voice and digital data). Within the set of possible digital data switching sub-domains, this International Standard is limited to a sub-set of these switching sub-domains with the following characteristics:

- connection-oriented
- circuit or cell-based packet switching
- point-to-point, and multi-point topology
- A connection in a call may represent either one or many data channels within the switching sub-domain.
- A CSTA Device can have multiple calls but the calls are totally unrelated to one another (i.e., they can not be conferenced, transferred, join, etc.).

Examples of these digital data switching sub-domains that support these characteristics are;

- T1
- ISO-Ethernet (TDM part of the protocol)
- ISDN
- Switched 56
- RSVP
- ATM (B-ISDN)

As a result, another attribute of a call is the Media Class (i.e., Voice or Data). Voice calls and the sub-set of digital data calls as described above use the same model for control and observation but with some additional unique characteristics for digital data. The following is the list of characteristics:

1. *Media Class* - Describes the type of call. It may consist of one or more of the following classes.
 - **Data** - These types of calls involve digital data calls (both circuit switched and packet switched). Devices that may be involved with these types of calls are digital computer interfaces and G4 facsimile machines.
 - **Image** - Digital data calls involving imaging, or high-speed, circuit-switched data in general. Devices that may be involved with these types of calls include digital video telephones and CODECs.
 - **Voice** - Devices in this class are used to make speech calls. This class includes standard telephones.
 - **Audio** - 3.1 KHz audio. Devices in this class are used to make audio calls excluding speech calls. It includes G3 fax and facsimile machines.
 - **Other** - A class comprising devices not in the Data, Image, Audio or Voice classes.
2. *Connection Rate* - This characteristic reflects the amount of bandwidth which is needed or allocated for a digital data call. This characteristic is specified in kilobits per second. This characteristic also represents the amount of bandwidth for both directions of data flow. The computing function can learn about a switching function's supported rates through the capability exchange services.
3. *Bit Rate* - This characteristic specifies if the bit rate is or needs to be a constant rate (i.e., dedicated bandwidth and constant rate of media stream delivery) or variable rate. A constant rate is used for media streams like audio or video. A variable rate is used for media stream like computer-generated data transfer.
4. *Delay Tolerance* - This characteristic specifies the maximum amount of media stream delivery delay that will be tolerated for the call. If the bit rate is constant, then this value will indicate the actual amount of media stream delivery delay for the life of the call. Where as if the bit rate is variable, it will be the maximum delay allowed during the life of the call.

5. *Switching Function Call Control Information Elements* - This characteristic provides a mechanism which enables the use of the switching function private call control information elements to be used during the life of a digital data call (e.g., elements used during call setup). The format, meaning and behaviour of these information elements are specific to the given switching function. This International Standard allows the following types of switching function private call control information elements:

- *ISDN* - All information elements associated with call control from the ITU Q.931 standard.
- *ATM* - All information elements associated with call control from the ITU B-ISDN Q.2931 standard.
- *RSVP* - All information elements associated with the call control from the IETF RSVP functional specification.
- *ISO-Ethernet (TDM part of the protocol)* - All information elements associated with call control from the 802.9 standard (a modified version of Q.933).
- *Private ISDN* - All information elements associated with call control and supplementary service control from ISO/IEC 11572 and ISO/IEC 11582.
- *Other* - All information elements associated with call control from the particular switching function specification.

For details on the specific information elements of these types, see the appropriate standards or switching function specific documentation. If a given information element overlaps with an attributes or characteristic that is defined by this International Standard, the information element should not be used and the attribute or characteristic in this International Standard should be used. If both are supplied, then they shall contain the same value. In addition, any information element that deals with device addressing shall not be used (e.g., calling party number, calling party subaddress). The CSTA Device addressing in this International Standard shall be used.

Once these characteristics are established for a digital data call, they can not be changed for the life of the call (i.e., they are static). These characteristics apply to all connections of the call.

The combination of these characteristics represents the *quality of service* associated with a given digital data call. The meaning of the quality of service which respect to characteristic combination is switching function specific.

6.1.4.7 Call Linkage

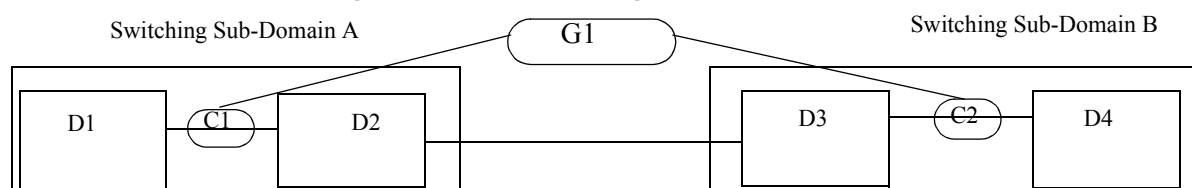
Although each CSTA call is independent of any other call (e.g., uniquely identified via its own call identifier), some CSTA calls can be related to other calls that exist in the same switching sub-domain or in different switching sub-domains.

6.1.4.7.1 Global Calls

Figure 6-16 shows how a call in one switching sub-domain is related to a call in another switching sub-domain because both calls are part of the same “end-to-end” or “global” call. Even though there is a CSTA call in each switching sub-domain (C1 and C2), both calls are part of the same “end-to-end” or “global” call identified as G1 in the figure.

The call linkage feature provides a reference (globalCallData) that can be used to represent the global call.

Figure 6-16 Illustration of a “global” or “end-to-end” call



legend:

- Device D1 in sub-domain A called Device D4 in sub-domain B.
- CSTA Call C1 represents a call in switching sub-domain A between calling device (D1) and an outgoing trunk (D2).
- CSTA Call C2 represents a call in switching sub-domain B between incoming trunk (D3) and a called device (D4).
- Global Call G1 represents the end-to-end view of the call.

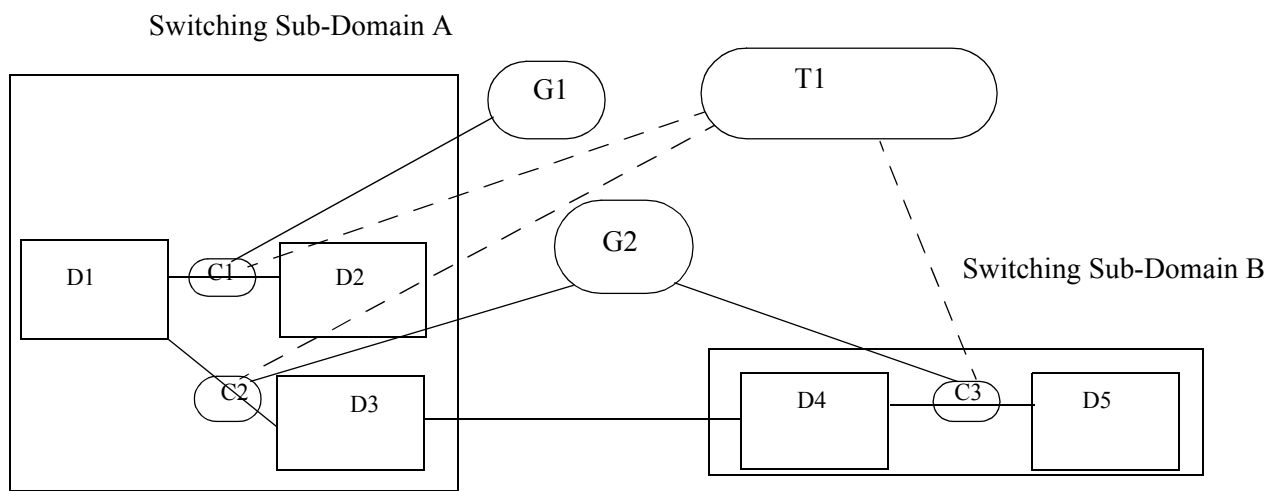
6.1.4.7.2 Call Threads

Different calls (CSTA calls or global calls) can be associated because they are part of the same or part of a related telephony process or call thread.

For example, as part of a CSTA Consultation Call service, a call that is placed on hold is associated or linked to the newly created consulted call.

Figure 6-17 illustrates the relationship between three CSTA calls (C1, C2, C3), two global calls (G1 and G2), and one call thread (T1) as the result of when device D1 in switching sub-domain A consults with device D5 in switching sub-domain B.

Figure 6-17 Illustration of a Call Thread



Legend:

- Device D1 has a call with device D2 and has consulted with device D5 in switching sub-domain B.
- (D3 is an outgoing trunk and D4 is an incoming trunk).
- There are two “end-to-end” or “global calls”: G1 involves devices D1 and D2. G2 involves devices D1, D3, D4, and D5.
- There is one thread identified as T1. This thread involves calls C1, C2, and C3.

The call linkage feature provides a reference (threadData) that can be used to associate different CSTA calls in a call thread. This information may be used by applications for correlating calls in multiple switching sub-domains, for correlating calls for charging/accounting purposes, or for call reporting purposes, etc.

6.1.4.7.3 Call Linkage Data

If the switching function supports the call linkage feature, it maintains call linkage data for every call and provides it on call control events (where specified) via the callLinkageData parameter (see 12.2.5, “CallLinkageData”, on page 87). Call linkage data can also be obtained via the Snapshot Call service.

The call linkage data consists of two components each representing a different relationship of the call to other calls. The globalCallData component is always provided if the switch supports the call linkage feature. The threadData component is only provided if the switching function also supports the thread linkage feature (as indicated by the capability exchange services). The two components are:

1. globalCallData - This component represents a “global” or “end-to-end” view of the call. The global call may involve one or more endpoints. The endpoints may reside in the same switching sub-domain or in multiple sub-domains. If the call resides in multiple switching sub-domains, there may be multiple switching sub-domain (CSTA) calls involved in the global call, all referenced by the same globalCallData. When a new call is created, a switching function shall either:
 - create new globalCallData for that call if the new call is not part of an existing global call
 - use the existing call’s globalCallData for the new call if the new call is part of an existing global call. This occurs when call arrives from another switching sub-domain and globalCallData is provided with the call.

2. **threadData** - This component represents a call thread - the set of calls that are associated because they are part of the same or a related telephony process. The call thread may involve multiple (CSTA and global) calls. When a new call is created, a switching function shall either:
- create new threadData for that call if there is no linkage to another call.
 - use existing threadData from a linked call for the new call (i.e., the call inherits the threadData from another call). The threadData can be inherited from a call from the same switching sub-domain (as part of a CSTA Consultation Call service, for example) or can be inherited from a call from another switching sub-domain (if provided with the call).
 - When two calls are conferenced or transferred, the following inheritance rules apply:
 - if only one of the calls was an incoming call (where an incoming call is defined as a call that was not originated at this device) at the conferencing/transferring device, the threadData for the resulting call shall be inherited from the incoming call. The threadData associated with the other call shall be discarded.
 - if both of the calls were incoming calls at the conferencing/transferring device, the threadData for the resulting call shall be inherited from the call that arrived first at the conferencing/transferring device. The threadData associated with the other call shall be discarded.
 - if both calls originated at the conferencing/transferring device, the threadData for the resulting call shall be inherited from the first call that originated at the conferencing/transferring device. The threadData associated with the other call shall be discarded.

Refer to 12.2.5, “CallLinkageData”, on page 87 for more information on the CallLinkageData parameter type.

6.1.4.7.4 Synchronization of Call Linkage Data

When callLinkageData associated with a call is updated due to a telephony feature, callLinkageData shall be synchronized between calls in a switching sub-domain or between calls in different switching sub-domains. For example, if a call is sent from an originating switching sub-domain to a terminating switching sub-domain, and then the callLinkageData is changed in the terminating switching sub-domain because of a conference, then the terminating switching sub-domain shall notify the originating switching sub-domain that its callLinkageData has changed so that the originating switching sub-domain can associate the updated callLinkageData with its call. (Annex B defines one possible mechanism that could be used for transporting the call linkage data through an external network.)

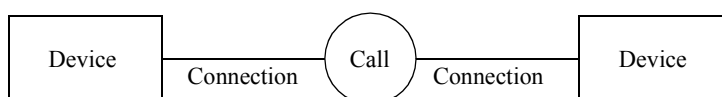
The following CSTA events indicate when the callLinkageData associated with a call has been updated:

- The CSTA Conferenced and Transferred events are used to indicate that the callLinkageData has changed as a result of a conference or transfer feature.
- Otherwise, the CSTA Call Information event is used when the callLinkageData has been updated, in the same switching sub-domain that is reporting the Call Information event or in a different switching sub-domain. For example, for a call between two switching sub-domains, the Call Information event can be used to indicate that the callLinkageData has changed as a result of a feature in a *different* switching sub-domain (the mechanism described in Annex B is one way to transport the callLinkageData over the external network).

6.1.5 Connection

A *connection* is a relationship in a switching sub-domain between a CSTA Device, and a call in which that CSTA Device is involved. This connection relationship can be both observed and manipulated. Figure 6-18 illustrates the relationship between calls, devices, and connections.

Figure 6-18 Relationship between Calls, Devices, and Connections



Observation and manipulation of these connections are the basis for call control services (such as Clear Connection, Answer Call, etc.). Connections are CSTA Objects that have the following attributes:

1. *Connection Identifier* - Each connection that can be observed and/or controlled shall be referenced across the service boundary. To accomplish this, each connection is assigned a unique identifier by the switching function. This identifier is comprised of a Device Identifier and a Call Identifier. For a call, there are as many Connection Identifiers as there are associated devices, and for a device there are as many Connection Identifiers as there are associated calls. The Connection Identifier is unique within a sub-domain and over a single service boundary. It is provided by the switching function when either a new call is created or a new device becomes involved in a call. A Connection Identifier can change as a result of some operations (e.g., a transfer or conference) and in these cases the switching function presents the computing function with the appropriate information to transit from the old identifiers to the new. The Device Identifier used in the Connection Identifier may be either static or dynamically-assigned by the switching function.

As provided by the switching function to the computing function, a Connection Identifier will always include both a Device Identifier and a Call Identifier (unless otherwise noted in the specification of a particular CSTA event's parameters). Computing functions wanting to correlate event reports which associate devices connected together in a call can use the Call Identifier to do this correlation. The definitions of a Connection Identifier and those identifiers that it comprises (Call and Device Identifiers) restrict computing functions from fabricating Connection Identifiers.

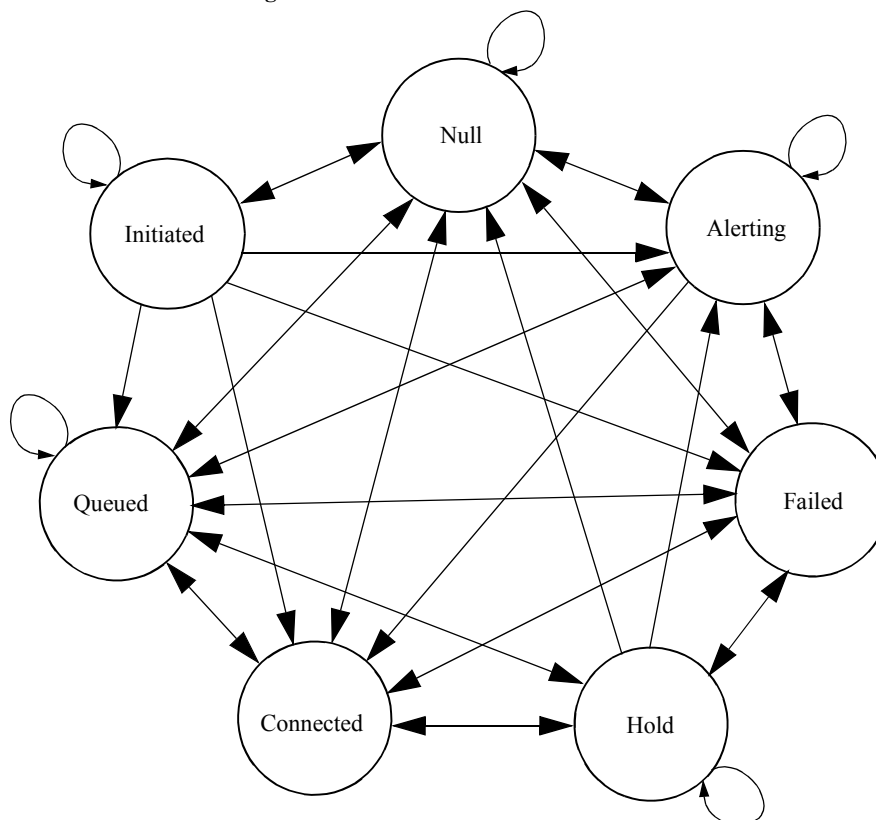
As provided by the computing function to the switching function, a Connection Identifier shall be one which was originally provided by the switching function. An exception to this rule is where either a deviceID only or a callID only Connection Identifier is used in a specific service (as indicated by the capability exchange services). If a Connection Identifier, provided by the computing function, includes only a Device Identifier, then that Device Identifier shall be a static Device Identifier. These conditions ensure that it is possible to use only a Device Identifier (without a Call Identifier) or a Call Identifier (without a Device Identifier) to provide a Connection Identifier in certain specified circumstances.

For additional details regarding Connection Identifiers, including Connection Identifier formats and specific functional requirements, see 12.3.9, "ConnectionID", on page 114.

2. *Media Stream Flow Direction* - This attribute is the direction or directions in which the media stream can be transmitted on the given connection. The following are the types of directions that can be associated with a connection:
 - *Transmit* - Media stream data can only be transmitted on the connection by the associated device.
 - *Receive* - Media stream data can only be received on the connection by the associated device.
 - *Transmit and Receive* - Media stream data can be transmitted and received on the connection by the associated device.
3. *Media Stream Channels* - A channel is a path of communications between devices within a network. Channels are created to transmit/receive the media stream when the associated connection is created for the device in the call. The correlation of a channel to an actual media stream communications path/channel within the switching function is switching function specific. The switching function may represent a channel as a group of actual media stream communication paths. A device's connection represents a channel or set of channels on which the media stream associated with the call is to be transmitted and received. The number of channels per connection is switching function and device specific (the capability exchange services may be used to determine the value). A digital data connection can use one or more channels. In addition, there can be multiple media stream types associated with a given connection as well as the associated channels. The attachment of media services is to a connection and its associated channels as a whole. The switching function is then responsible for attaching the Media services to the appropriate channels.
4. *Connection State* - A connection state involves a single call/device relationship. When a call is present at a device, the connection representing that call at that device will transit through various stages. State transitions are observed by the computing function through event reports. The transition from one state to the next is

caused by either a manual user stimulus or a CSTA service initiated across the service boundary. Connection states may also be reported by Snapshots on either calls or devices.

Figure 6-19 Connection State Model



The following are the connection state definitions:

- *Alerting* - A state in which an attempt is being made to connect a call to a device. There are three distinct modes where a connection may be in the alerting state:
 - *Offered* - In this mode, the call is in a pre-delivery state at the target device. The opportunity exists for a computing function to issue one of a set of supported services (e.g., Accept Call, Clear Connection (“reject”), Deflect Call) or an ISDN device to accept or reject the call. From the calling side perspective, the call is not delivered at the called device. As a consequence, delivery information such as Ringback indication and/or Network signalling is not provided. For example, the device makes no ringing sounds while in the Offered mode of the Alerting state. The Offered mode is indicated by an Offered event.
 - *Ringing* - In this mode, the call is being presented for the purpose of having the device connect to the call and the user is made aware that the call is being delivered at the device. The Ringing mode is indicated by a Delivered event with a cause code other than “Entering Distribution”. The actual device activity to notify the user (e.g., ringing) is reported through the physical device feature events.
 - *Entering Distribution* - In this mode, a call is being delivered to a distribution device. The Entering Distribution mode is indicated by a Delivered event with a cause code of “Entering Distribution”.
- *Connected* - A state in which a device is actively participating in a call. This state includes logical participation in a call as well as physical participation.
- *Failed* - A state in which call progression has been aborted. This state generally results when a device tries to become Connected to a call or a call tries to become Connected to a device and the attempt fails. The Failed state can result from failure to connect the calling device and call, failure to connect the called device and call, failure to create the call, failure when the call ends and other reasons. Refer to 6.8.2, “Connection Failure”, on page 55 for more information.

- *Hold* - A state in which a device is inactive participating in a call. This state includes logical participation in a call while physical participation is suspended.
- *Initiated* - A state in which a device is requesting a service or in the process of dialling the necessary digit sequence to initiate a call to another device. The connection enters this state when the device goes off-hook (e.g., receiving dialtone) or the device is being prompted to go off-hook as a result of some service being initiated for the device.
- *Null* - A state in which there is no relationship between a call and device.
- *Queued* - A state in which call progression is suspended or made inactive while awaiting some form of action. Examples of situations in which a connection might transit to the Queued state include (among others) the following:
 - A call is parked at a device.
 - A call is queued at a distribution mechanism, waiting for an agent to become available.
 - A call is camped on to a device.
 - An appearance of a shared bridged device configuration is inactive with respect to the call.

Table D-1 on page 636 provides an example to illustrate each transition which is illustrated in Figure 6-19, "Connection State Model".¹

6.1.5.1 Call Event Reports

The Connection state model provides an abstract view of actual states and events that are communicated via underlying signalling systems. This abstract view is introduced to provide a language for describing CSTA Event Reports, states and Functional descriptions. Because of the topology of the Switching Function, the signals that report events and state changes have definite sources. Providing a telecommunications object (the Connection) that can be associated with the source of these signals helps when explaining the meaning of events and the operation of CSTA (and other) telecommunications services.

Note that on a typical ISDN access to a network there exists a distributed state machine. One part of this distributed state machine resides in the ISDN device. Another part resides on the other side of the ISDN access. There is another similar distributed-state access machine that resides across the ISDN network at a similar device. Using this concept, a call can be modeled as a collection of Connection state machines communicating with one another using signalling. When this communication occurs, a CSTA Event Report can be generated. In the following figure, this concept of communication between two state machines is illustrated for the case of establishing a simple call. Additionally, on each side of the figure the ISDN call states are indicated.

1. For an explanation of the Initial and Final State diagrams' nomenclature used the Table D-1, refer to Clause 11, "Template Descriptions", beginning on page 80.