

JT-Q1228

IN(インテリジェントネットワーク)
網間インタフェース能力セット2
- プロトコル -

IN(Intelligent network)
Inter-Network Interface Capability Set2
-Protocol-

第1版

1999年4月22日制定

社団法人
情報通信技術委員会

THE TELECOMMUNICATION TECHNOLOGY COMMITTEE

本書は、（社）情報通信技術委員会が著作権を保有しています。

内容の一部又は全部を（社）情報通信技術委員会の許諾を得ることなく複製、転載、改変、転用及びネットワーク上での送信、配布を行うことを禁止します。

<参考>

1. 国際勧告との関連

本標準は1997年9月に開催されたITU-T SG11 会合において承認されたITU-T勧告Q.1228に準拠している。

2. 上記国際勧告等に対する追加事項等

2.1 オプション選択項目

なし

2.2 ナショナルマター選択項目

なし

2.3 その他

- (1) 本標準は、上記ITU-T勧告に対して先行して規定する事項を含んではいない。
- (2) 本標準は、上記ITU-T勧告に対して、以下の原則により削除を示している。
 - (a) ITU-T勧告で規定しているが、本標準では規定しない章、節および項は、目次および本文中に章、節および項の番号とタイトルのみを示し、#で表示している。なお、上記ITU-T勧告で規定しているが、本標準では規定しない記述で、理解を助けるために本文中に残しているものは#で表示する。
 - (b) ITU-T勧告で規定しているが、本標準では規定しない章、節および項内の記述は、特に削除を示す表示を付与することなく削除している。
- (3) 本標準は、インテリジェントネットワークのサービス制御機能 - サービスデータ機能間およびサービスデータ機能相互間のインタフェースを標準仕様として規定するためのものであるため、上記ITU-T勧告に対して、この目的のために不必要である、以下の項目についての記述を削除している。

2.2節	略語と頭字語
3.1.2.1節	「SCF - 外部SRF」中継による通信の例
4.1節	データ型
4.2節	エラー型
4.3節	オペレーションコード
4.4節	エラーコード
5章	SSF / CCF - SCFインタフェース
6章	SCF - SRFインタフェース
9章	SCF / SCFインタフェース
10章	SCF / CUSFインタフェース
11章	SSFアプリケーションエンティティ手順
12.4節	部分的なSCF管理エンティティ(SCME)状態遷移図
12.5.1項	SSF / SRF関連状態(SCSM - SSF / SRF)
12.5.3項	SCF関連状態
12.5.4項	CUSF関連状態(SCSM - CUSF)

1 2 . 5 . 5 項	U S I _ S C F F S M
1 3 章	S R F アプリケーションエンティティ手順
1 5 章	C U S F アプリケーションエンティティ手順
1 6 . 1 . 2 項	Canceled
1 6 . 1 . 3 項	CancelFailed
1 6 . 1 . 5 項	ETCFailed
1 6 . 1 . 7 項	ImproperCallerResponse
1 6 . 1 . 8 項	MissingCustomerRecord
1 6 . 1 . 9 項	MissingParameter
1 6 . 1 . 1 1 項	ParameterOutOfRange
1 6 . 1 . 1 3 項	RequestedInfoError
1 6 . 1 . 1 4 項	ScfReferral
1 6 . 1 . 1 5 . 4 項	S C F - > S C F オペレーション
1 6 . 1 . 1 8 項	SystemFailure
1 6 . 1 . 1 9 項	TaskRefused
1 6 . 1 . 2 0 項	UnavailableResource
1 6 . 1 . 2 1 項	UnexpectedComponentSequence
1 6 . 1 . 2 2 項	UnexpectedDataValue
1 6 . 1 . 2 3 項	UnexpectedParameter
1 6 . 1 . 2 4 項	UnknownLegID
1 6 . 1 . 2 5 項	UnknownResource
1 6 . 1 . 2 7 項	ChainingRefused
1 6 . 1 . 2 9 項	ScfBindFailure
1 6 . 1 . 3 0 項	ScfTaskRefused
1 6 . 2 節	エンティティ関連エラー手順
1 7 . 1 節	ActivateServiceFilterin 手順
1 7 . 2 節	ActivationReceivedAndAuthorized 手順
1 7 . 3 節	ActivityTest 手順
1 7 . 5 節	AnalysedInformation 手順
1 7 . 6 節	AnalyseInformation 手順
1 7 . 7 節	ApplyCharging 手順
1 7 . 8 節	ApplyChargingReport 手順
1 7 . 9 節	AssistRequestInstructions 手順
1 7 . 1 0 節	AssociationReleaseRequested 手順
1 7 . 1 1 節	AuthorizeTermination 手順
1 7 . 1 2 節	CallGap 手順
1 7 . 1 3 節	CallInformationReport 手順
1 7 . 1 4 節	CallInformationRequest 手順
1 7 . 1 5 節	Cancel 手順

1 7 . 1 6 節	CancelStatusReportRequest 手順
1 7 . 1 8 節	ChainedConfirmedNotificationProvided 手順
1 7 . 1 9 節	ChainedConfirmedReportChargingInformation 手順
1 7 . 2 0 節	ChainedEstablishChargingRecord 手順
1 7 . 2 2 節	ChainedHandlingInformationRequest 手順
1 7 . 2 3 節	ChainedHandlingInformationResult 手順
1 7 . 2 5 節	ChainedNetworkCapability 手順
1 7 . 2 6 節	ChainedNotificationProvided 手順
1 7 . 2 7 節	ChainedReportChargingInformation 手順
1 7 . 2 8 節	ChainedProvideUserInformation 手順
1 7 . 3 0 節	ChainedRequestNotification 手順
1 7 . 3 2 節	CollectedInformation 手順
1 7 . 3 3 節	CollectInformation 手順
1 7 . 3 4 節	ComponentReceived 手順
1 7 . 3 5 節	ConfirmedNotificationProvided 手順
1 7 . 3 6 節	ConfirmedReportChargingInformation 手順
1 7 . 3 7 節	Connect 手順
1 7 . 3 8 節	ConnectToResource 手順
1 7 . 3 9 節	Continue 手順
1 7 . 4 0 節	ContinueWithArgument 手順
1 7 . 4 2 節	CreateCallSegmentAssociation 手順
1 7 . 4 5 節	DisconnectForwardConnection 手順
1 7 . 4 6 節	DisconnectForwardConnectionWithArgument 手順
1 7 . 4 7 節	DisconnectLeg 手順
1 7 . 5 1 節	EntityReleased 手順
1 7 . 5 2 節	EstablishChargingRecord 手順
1 7 . 5 3 節	EstablishTemporaryConnection 手順
1 7 . 5 4 節	EventNotificationCharging 手順
1 7 . 5 5 節	EventReportBCSM 手順
1 7 . 5 6 節	EventReportFacility 手順
1 7 . 5 8 節	FacilitySelectedAndAvailable 手順
1 7 . 5 9 節	FurnishChargingInformation 手順
1 7 . 6 0 節	HandlingInformationRequest 手順
1 7 . 6 1 節	HandlingInformationResult 手順
1 7 . 6 2 節	HoldCallInNetwork 手順
1 7 . 6 4 節	InitialDP 手順
1 7 . 6 5 節	InitiateAssociation 手順
1 7 . 6 6 節	InitiateCallAttempt 手順
1 7 . 6 7 節	ManageTriggerData 手順

1 7 . 6 8 節	MergeCallSegments 手順
1 7 . 7 0 節	MoveCallSegments 手順
1 7 . 7 1 節	MoveLeg 手順
1 7 . 7 2 節	NetworkCapability 手順
1 7 . 7 3 節	NotificationProvided 手順
1 7 . 7 4 節	OAbandon 手順
1 7 . 7 5 節	OAnswer 手順
1 7 . 7 6 節	OCalledPartyBusy 手順
1 7 . 7 7 節	ODisconnect 手順
1 7 . 7 8 節	OMidCall 手順
1 7 . 7 9 節	ONoAnswer 手順
1 7 . 8 0 節	OriginationAttempt 手順
1 7 . 8 1 節	OriginationAttemptAuthorized 手順
1 7 . 8 2 節	OSuspended 手順
1 7 . 8 3 節	PlayAnnouncement 手順
1 7 . 8 4 節	PromptAndCollectUserInfo 手順
1 7 . 8 5 節	PromptAndReceiveMessage 手順
1 7 . 8 6 節	ProvideUserInfo 手順
1 7 . 8 7 節	Reconnect 手順
1 7 . 8 8 節	ReleaseAssociation 手順
1 7 . 8 9 節	ReleaseCall 手順
1 7 . 9 1 節	ReportChargingInformation 手順
1 7 . 9 2 節	ReportUTSI 手順
1 7 . 9 3 節	RequestCurrentStatusReport 手順
1 7 . 9 4 節	RequestEveryStatusChangeReport 手順
1 7 . 9 5 節	RequestFirstStatusMatchReport 手順
1 7 . 9 6 節	RequestNotification 手順
1 7 . 9 7 節	RequestNotificationChargingEvent 手順
1 7 . 9 8 節	RequestReportBCSMEEvent 手順
1 7 . 9 9 節	RequestReportBCUSMEEvent 手順
1 7 . 1 0 0 節	RequestReportFacilityEvent 手順
1 7 . 1 0 1 節	RequestReportUTSI 手順
1 7 . 1 0 3 節	ResetTimer 手順
1 7 . 1 0 4 節	RouteSelectFailure 手順
1 7 . 1 0 5 節	SCFBind 手順
1 7 . 1 0 6 節	scfBind 手順 (連鎖の場合)
1 7 . 1 0 7 節	SCFUnBind 手順
1 7 . 1 0 8 節	scfUnBind 手順 (連鎖の場合)
1 7 . 1 0 9 節	ScriptClose 手順

17.110 節	ScriptEvent 手順
17.111 節	ScriptInformation 手順
17.112 節	ScriptRun 手順
17.114 節	SelectFacility 手順
17.115 節	SelectRoute 手順
17.116 節	SendChargingInformation 手順
17.117 節	SendComponent 手順
17.118 節	SendFacilityInformation 手順
17.119 節	SendSTUI 手順
17.120 節	ServiceFilteringResponse 手順
17.121 節	SpecializedResourceReport 手順
17.122 節	SplitLeg 手順
17.123 節	StatusReport 手順
17.124 節	TAnswer 手順
17.125 節	TBusy 手順
17.126 節	TDisconnect 手順
17.127 節	TerminationAttempt 手順
17.128 節	TermAttemptAuthorized 手順
17.129 節	TMidCall 手順
17.130 節	TNoAnswer 手順
17.131 節	TSuspended 手順
18.1.2 節	S S F - S C F インタフェース
18.1.3 節	S C F - S R F インタフェース
18.1.4 節	S C F - C U S F インタフェース
18.1.5 節	S C F - S C F インタフェース
付属資料 A	I N A P の S D L 図
付録 I	拡張 A S N . 1
付録 III	S P K M アルゴリズムの例

- (4) 本標準において、TTC 特有に追加した記述部分については、* で表示する。また、本標準の参照において有用と考えられる用語、略語については、付属資料 B として追加している。

2.4 原勧告と章立ての構成比較表

上記国際勧告との章立ての構成の相違はない。

3. 改版の履歴

版数	制定日	改版内容
第 1 版	1999 年 4 月 22 日	制定

4．工業所有権

本標準に関わる「工業所有権の実施の権利に係る確認書」の提出状況は、TTCホームページでご覧になれます。

目次

1 . 序論	1	
2 . 総論	2	
2.1 規範となる参照先	2	
2.2 略語と頭字語	3	#
3 . 電気通信サービスのためのインタフェース標準	4	
3.1 総論	4	
3.1.1 定義方法論	4	
3.1.2 物理シナリオ例	4	
3.1.3 I N A P プロトコルアーキテクチャ	5	
3.1.4 I N A P アドレッシング	6	
3.1.5 T T C 標準 J T - Q 1 2 2 4 と本標準との関係	8	*
3.1.6 I N A P のために利用されるコンパチビリティメカニズム	9	
3.2 S A C F / M A C F 規則	10	
3.2.1 T C A C の反映	10	
3.2.2 オペレーションの直列 / 並列実行	10	
4 . 共通 I N C S - 2 型	12	
4.1 データ型	12	#
4.2 エラー型	12	#
4.3 オペレーションコード	12	#
4.4 エラーコード	12	#
4.5 クラス	12	
4.6 オブジェクト識別子	20	
5 . S S F - S C F インタフェース	25	#
6 . S S F - S R F インタフェース	25	#
7 . S C F - S D F インタフェース	26	
7.1 I N の S D F インタフェースにおける T T C 標準 J T - X 5 0 0 の再利用の導入	26	
7.1.1 T T C 標準 J T - X 5 0 0 の概念と I N との整合	26	
7.1.2 T T C 標準 J T - X 5 0 0 の限定されたサブセットの利用	26	
7.1.3 前提条件	27	
7.2 S D F 情報モデル	27	
7.2.1 情報フレームワーク	27	
7.2.2 基本アクセス制御	29	
7.2.3 属性コンテキスト	30	
7.2.4 属性定義	31	
7.3 S C F - S D F インタフェースプロトコル	32	
7.3.1 情報タイプおよび共通手順	32	
7.3.2 オペレーション	34	
7.3.3 Errors	38	
7.4 プロトコルの概略	39	
7.4.1 遠隔オペレーション	39	
7.4.2 ディレクトリの R O S オブジェクトおよびコントラクト	39	

7.4.3	D A P C o n t r a c t およびパッケージ	40
7.5	ディレクトリプロトコル抽象構文	42
7.5.1	抽象構文	42
7.5.2	ディレクトリアプリケーションコンテキスト	43
7.5.3	オペレーションコード	44
7.5.4	エラーコード	44
7.5.5	バージョンおよび拡張ルール	45
7.6	パフォーマンス	46
7.6.1	S C F によるパフォーマンス	46
7.6.2	S D F によるパフォーマンス	47
7.7	S C F - S D F インターフェースの A S N , 1 モジュール	48
7.7.1	IN-CS2-SDF-InformationFramework モジュール	48
7.7.2	IN-CS2-SDF-BasicAccessControl モジュール	50
7.7.3	IN-CS2-SCF-SDF-Operations モジュール	53
7.7.4	IN-CS2-SCF-SDF-Protocol モジュール	56
8	S D F - S D F インタフェース	60
8.1	I N T T C 標準 J T - X 5 0 0、D S P、D I S P サブセットへの導入	60
8.2	前提条件	60
8.3	I N T T C 標準 J T - X 5 0 0、D I S P サブセット	61
8.3.1	シャドウイング合意の規定	61
8.3.2	DSA Shadow Bind	61
8.3.3	IN-DSA Shadow Unbind	61
8.3.4	Coordinate Shadow Update	61
8.3.5	Update Shadow	62
8.3.6	Request Shadow Update	64
8.4	I N T T C 標準 J T - X . 5 0 0、D S P サブセット	64
8.4.1	情報タイプおよび共通手順	64
8.4.2	DSA Bind	69
8.4.3	IN DSA Unbind	69
8.4.4	連鎖オペレーション	69
8.4.5	連鎖エラー	70
8.5	プロトコルの概略	71
8.5.1	ROS オブジェクトおよびコントラクト	71
8.5.2	DSP コントラクトおよびパッケージ	71
8.5.3	DISP コントラクトおよびパッケージ	73
8.6	プロトコル抽象構文	74
8.6.1	DSP 抽象構文	74
8.6.2	DISP 抽象構文	74
8.6.3	ディレクトリシステムアプリケーションコンテキスト	75
8.6.4	ディレクトリシャドウアプリケーションコンテキスト	76
8.6.5	バージョンおよび拡張ルール	77
8.7	パフォーマンス	79
8.7.1	S D F によるパフォーマンス	79

8.7.2	シャドウ供給側によるパフォーマンス	80	
8.7.3	シャドウ消費側によるパフォーマンス	81	
8.8	SDF - SDF インタフェースのASN、1 モジュール	82	
8.8.1	IN-CS2-SDF-SDF-Protocol モジュール	82	
9	SCF - SCF インタフェース	87	#
10	SCF - CUSF インタフェース	87	#
11	SSF アプリケーションエンティティ手順	87	#
12	SCF アプリケーションエンティティ手順	88	
12.1	概要	88	
12.2	モデルとインタフェース	88	
12.3	SCF FSM と SLP / 保守機能間の相互関係	89	
12.4	部分的な SCF 管理エンティティ (SCME) 状態遷移図	91	#
12.5	SCF 呼状態モデル (SCSM)	91	
12.5.1	SSF / SRF 関連状態 (SCSM-SSF/SRF)	91	#
12.5.2	SDF 関連状態 (SCSM-SDF)	91	
12.5.3	SCF 関連状態	93	#
12.5.4	CUSF 関連状態 (SCSM-CUSF)	93	#
12.5.5	USI__SCF FSM	93	#
13	SRF アプリケーションエンティティ手順	93	#
14	SDF アプリケーションエンティティ手順	94	
14.1	概要	94	
14.2	モデルとインタフェース	94	
14.3	SDF FSM 構造	96	
14.4	SDF 状態遷移モデル	97	
14.4.1	SCF 関連状態に対する SDF 状態遷移モデル	97	
14.4.2	SDF 関連状態に対する SDF 状態遷移モデル	98	
15	CUSF アプリケーションエンティティ手順	115	#
16	エラー手順	116	
16.1	オペレーション関連エラー手順	116	
16.1.1	AttributeError	116	
16.1.2	Canceled	117	#
16.1.3	CancelFailed	117	#
16.1.4	DSAReferral	117	
16.1.5	ETCFailed	118	#
16.1.6	Execution エラー	118	
16.1.7	ImproperCallerResponse	120	#
16.1.8	MissingCustomerRecord	120	#
16.1.9	MissingParameter	120	#
16.1.10	NameError	120	
16.1.11	ParameterOutOfRange	122	#
16.1.12	Referral	122	
16.1.13	RequestedInfoError	123	#
16.1.14	ScfReferral	123	#

16.1.15	Security	123	
16.1.16	Service	125	
16.1.17	Shadow	126	
16.1.18	SystemFailure	129	#
16.1.19	TaskRefused	129	#
16.1.20	UnavailableResource	129	#
16.1.21	UnexpectedComponentSequence	129	#
16.1.22	UnexpectedDataValue	129	#
16.1.23	UnexpectedParameter	129	#
16.1.24	UnknownLegID	129	#
16.1.25	UnknownResource	129	#
16.1.26	Update	129	
16.1.27	chainingRefused	130	#
16.1.28	DirectoryBindError	130	
16.1.29	ScfBindFailure	132	#
16.1.30	ScfTaskRefused	132	#
16.2	エンティティ関連エラー手順	132	#
17	詳細オペレーション手順	132	
17.1	ActiveServiceFiltering 手順	132	#
17.2	ActivationReceivedAndAuthorized 手順	132	#
17.3	ActivityTest 手順	132	#
17.4	AddEntry 手順	132	
17.4.1	一般記述	132	
17.4.2	起動側エンティティ(S C F)	133	
17.4.3	応答側エンティティ(S D F)	133	
17.5	AnalysedInformation 手順	134	#
17.6	AnalyseInformation 手順	134	#
17.7	ApplyCharging 手順	134	#
17.8	ApplyChargingReport 手順	134	#
17.9	AssistRequestInstructions 手順	134	#
17.10	AssociationReleaseRequested 手順	134	#
17.11	AuthorizeTermination 手順	134	#
17.12	CallGap 手順	134	#
17.13	CallInformationReport 手順	134	#
17.14	CallInformationRequest 手順	134	#
17.15	Cancel 手順	134	#
17.16	CancelStatusReportRequest 手順	134	#
17.17	chainedAddEntry 手順	134	
17.17.1	一般記述	134	
17.17.2	起動側エンティティ(S D F)	134	
17.17.3	応答側エンティティ(S D F)	135	
17.18	chainedConfirmaedNotificationProvided 手順	136	#
17.19	chainedConfirmedReportChargingInformation 手順	136	#

17.20	chainedEstablishChargingRecord 手順	136	#
17.21	chainedExecute 手順	136	
17.21.1	一般記述	136	
17.21.2	起動側エンティティ (S D F)	136	
17.21.3	応答側エンティティ (S D F)	137	
17.22	chainedHandlingInformationRequest 手順	137	#
17.23	chainedHandlingInformationResult 手順	137	#
17.24	chainedModifyEntry 手順	137	
17.24.1	一般記述	137	
17.24.2	起動側エンティティ (S D F)	138	
17.24.3	応答側エンティティ (S D F)	138	
17.25	chainedNetworkCapability 手順	139	#
17.26	chainedNotificationProvided 手順	139	#
17.27	chainedReportChargingInformation 手順	139	#
17.28	chainedProvideUserInformation 手順	139	#
17.29	chainedRemoveEntry 手順	139	
17.29.1	一般記述	139	
17.29.2	起動側エンティティ (S D F)	139	
17.29.3	応答側エンティティ (S D F)	140	
17.30	chainedRequestNotification 手順	141	#
17.31	chainedSearch 手順	141	
17.31.1	一般記述	141	
17.31.2	起動側エンティティ (S D F)	141	
17.31.3	応答側エンティティ (S D F)	142	
17.32	collectedInformation 手順	142	#
17.33	collectInformation 手順	142	#
17.34	componentReceived 手順	142	#
17.35	confirmedNotificationProvided 手順	142	#
17.36	confirmedReportChargingInformation 手順	142	#
17.37	connect 手順	142	#
17.38	connectToResouce 手順	142	#
17.39	continue 手順	142	#
17.40	continueWithArgument 手順	142	#
17.41	CoordinateShadowUpdate 手順	142	
17.41.1	一般記述	142	
17.41.2	供給側エンティティ (S D F)	143	
17.41.3	消費側エンティティ (S D F)	144	
17.42	CreateCallSegmentAssociation 手順	145	#
17.43	in-directoryBind 手順	145	
17.43.1	一般記述	145	
17.43.2	起動側エンティティ (S C F)	145	
17.43.3	応答側エンティティ (S D F)	146	
17.44	in-directoryUnbind 手順	146	

17.44.1	一般記述	146	
17.44.2	起動側エンティティ(S C F)	147	
17.44.3	応答側エンティティ(S D F)	147	
17.45	DisconnectForwardConnection 手順	147	#
17.46	DisconnectForwardConnectionWithArgument 手順	147	#
17.47	DisconnectLog 手順	147	#
17.48	dSABind 手順	147	
17.48.1	一般記述	147	
17.48.2	起動側エンティティ(S D F)	148	
17.48.3	応答側エンティティ(S D F)	148	
17.49	DSAShadowBind 手順	149	
17.49.1	一般記述	149	
17.49.2	供給側エンティティ(S D F)	149	
17.49.3	消費側エンティティ(S D F)	151	
17.50	in-DSAShadowUnbind 手順	152	
17.50.1	一般記述	152	
17.50.2	供給側エンティティ(S D F)	153	
17.50.3	消費側エンティティ(S D F)	153	
17.51	EntityReleased 手順	154	#
17.52	EstablishChargingRecord 手順	154	#
17.53	EstablishTemporaryConnection 手順	154	#
17.54	EventNotificationCharging 手順	154	#
17.55	EventReportBCSM 手順	154	#
17.56	EventReportFacility 手順	154	#
17.57	Execute 手順	154	
17.57.1	一般記述	154	
17.57.2	起動側エンティティ(S C F)	156	
17.57.3	応答側エンティティ(S D F)	156	
17.58	FacilitySelectedAndAvailable 手順	157	#
17.59	FurnishChargingInformation 手順	157	#
17.60	HandlingInformationRequest 手順	157	#
17.61	HandlingInformationResult 手順	158	#
17.62	HoldCallInNetwork 手順	158	#
17.63	in-DSAUnbind 手順	158	
17.63.1	一般記述	158	
17.63.2	起動側エンティティ(S D F)	158	
17.63.3	応答側エンティティ(S D F)	158	
17.64	InitialDP 手順	158	#
17.65	InitiateAssociation 手順	158	#
17.66	InitiateCallAttempt 手順	159	#
17.67	ManageTriggerData 手順	159	#
17.68	MergeCallSegments 手順	159	#
17.69	ModifyEntry 手順	159	

17.69.1	一般記述	159	
17.69.2	起動側エンティティ(S C F)	159	
17.69.3	応答側エンティティ(S D F)	160	
17.70	MoveCallSegments 手順	160	#
17.71	MoveLeg 手順	160	#
17.72	NetworkCapability 手順	160	#
17.73	NotificationProvided 手順	160	#
17.74	OAbandon 手順	160	#
17.75	OAnswer 手順	160	#
17.76	OCalledPartyBusy 手順	160	#
17.77	ODisconnect 手順	160	#
17.78	OMidCall 手順	160	#
17.79	ONoAnswer 手順	160	#
17.80	OriginationAttempt 手順	160	#
17.81	OriginationAttemptAuthorized 手順	160	#
17.82	OSuspended 手順	160	#
17.83	PlayAnnouncement 手順	160	#
17.84	PromptAndCollectUserInformation 手順	161	#
17.85	PromptAndReceiveMessage 手順	161	#
17.86	ProvideUserInformation 手順	161	#
17.87	Reconnect 手順	161	#
17.88	ReleaseAssociation 手順	161	#
17.89	ReleaseCall 手順	161	#
17.90	RemoveEntry 手順	161	
17.90.1	一般記述	161	
17.90.2	起動側エンティティ(S C F)	161	
17.90.3	応答側エンティティ(S D F)	162	
17.91	ReportChargingInformation 手順	162	#
17.92	ReportUTSI 手順	162	#
17.93	RequestCurrentStatusReport 手順	162	#
17.94	RequestEveryStatusChangeReport 手順	162	#
17.95	RequestFirstStatusMatchReport 手順	162	#
17.96	RequestNotification 手順	162	#
17.97	RequestNotificationChargingEvent 手順	162	#
17.98	RequestReportBCSMEEvent 手順	162	#
17.99	RequestReportBCUSMEEvent 手順	162	#
17.100	RequestReportFacilityEvent 手順	162	#
17.101	RequestReportUTSI 手順	162	#
17.102	RequestShadowUpdate 手順	162	
17.102.1	一般記述	162	
17.102.2	供給側エンティティ(S D F)	163	
17.102.3	消費側エンティティ(S D F)	163	
17.103	ResetTimer 手順	164	#

17.104	RouteSelectFailure 手順	164	#
17.105	SCFBind 手順	164	#
17.106	scfBind 手順 (連鎖の場合)	164	#
17.107	SCFUnBind 手順	164	#
17.108	scfUnBind 手順 (連鎖の場合)	164	#
17.109	ScriptClose 手順	164	#
17.110	ScriptEvent 手順	164	#
17.111	ScriptInformation 手順	164	#
17.112	ScriptRun 手順	165	#
17.113	Search 手順	165	
17.113.1	一般記述	165	
17.113.2	起動側エンティティ(SCF)	165	
17.113.3	応答側エンティティ(SDF)	165	
17.114	SelectFacility 手順	166	#
17.115	SelectRoute 手順	166	#
17.116	SendChargingInformation 手順	166	#
17.117	SendComponent 手順	166	#
17.118	SendFacilityInformation 手順	166	#
17.119	SendSTUI 手順	166	#
17.120	ServiceFilteringResponse 手順	166	#
17.121	SpecializedResourceReport 手順	166	#
17.122	SplitLeg 手順	166	#
17.123	StatusReport 手順	166	#
17.124	TAnswer 手順	166	#
17.125	TBusy 手順	166	#
17.126	TDisconnect 手順	166	#
17.127	TerminationAttempt 手順	166	#
17.128	TermAttemptAuthorized 手順	166	#
17.129	TMidCall 手順	166	#
17.130	TNoAnswer 手順	166	#
17.131	TSuspended 手順	166	#
17.132	UpdateShadow 手順	166	
17.132.1	一般記述	166	
17.132.2	供給側エンティティ(SDF)	167	
17.132.3	消費側エンティティ(SDF)	168	
18	下位レイヤから想定されるサービス	170	
18.1	TCから想定されるサービス	170	
18.1.1	共通手順	170	
18.1.2	SSF - SCFインタフェース	180	#
18.1.3	SCF - SRFインタフェース	180	#
18.1.4	SCF - CUSFインタフェース	180	#
18.1.5	SCF - SCFインタフェース	180	#
18.1.6	SCF - SDFインタフェース	180	

18.1.7	SDF - SDF インタフェース	184	
18.2	SCCP から想定されるサービス	186	
18.2.1	正常手順	186	
18.2.2	SCCP からのサービス機能	186	
19	IN の一般的なインタフェースセキュリティ	190	#
19.1	インタフェースセキュリティ要求条件	190	#
19.1.1	データ機密性	190	#
19.1.2	データ完全性とデータ生成源認証	190	#
19.1.3	鍵管理	190	#
19.2	手順とアルゴリズム	191	#
19.2.1	認証手順	191	#
19.2.2	SPKM のアルゴリズムとネゴシエーション	192	#
19.2.3	3 ウェイ相互認証	193	#
19.2.4	資格証明の割り当て	193	#
19.3	トークンへのセキュリティ情報フロー定義のマッピング	193	#
19.4	セキュリティ F S M 定義	193	#
19.4.1	2 ウェイ相互認証 F S M	193	#
19.4.2	3 ウェイ相互認証 F S M	196	#
付属資料 A	INAP の S D L 図	200	#
付属資料 B	インテリジェントネットワーク定義で使われた用語解説	200	*
1	用語と定義	200	*
2	略語	203	*
付録 I	拡張 ASN . 1	205	#
付録 II	データモデリング	205	
II.1	はじめに	205	
II.1.1	目的と範囲	205	
II.1.2	前提	205	
II.2	D I T スキーマ	206	
II.2.1	T T C 標準 J T - X 5 0 0 D I T	206	
II.2.2	オブジェクトクラス	208	
II.2.3	属性タイプ	209	
II.2.4	D I T 構造定義	210	
付録 III	SPKM アルゴリズムの例	212	#

1 . 序論

本標準は、インテリジェントネットワーク能力セット2 (I N C S - 2) のサポートのために必要とされる I N A P (インテリジェントネットワークアプリケーションプロトコル) を定義する。これは、 I N 機能モデルで定義されたように、以下の機能エンティティ (F E) の間の相互動作をサポートする。

- サービス制御機能 (S C F)
- サービスデータ機能 (S D F)

本標準の範囲は、統合サービスデジタル網 (I S D N) および公衆交換電気通信網 (P S T N) の双方に対する I N A P のさらなる進展である。

これは、装置開発者やネットワークオペレータに対して、本標準で定義されたインタフェースについての異なる製造者の装置間の相互接続、および網間インタフェースについてのネットワークオペレータ間の相互接続を保証するためのガイドとなるよう意図されている。

なお、本標準は S C F - S D F および S D F - S D F の相互関係に関わるインタフェースを対象として規定するものである。

*

*

2 . 総論

2.1 規範となる参照先

「以下のTTC標準および他のリファレンスは、本テキストから参照されることにより、本標準もしくは本標準への付録に対する規定が制定されている規定を含んでいる。」

- ITU - T勧告 X . 5 0 0 (1997) | ISO / IEC 9 5 9 4 - 1 : 1997、
Information technology - Open Systems Interconnection -
The Directory: Overview of Concepts, Models and Services .
- TTC標準 JT - X 5 0 0 (1994) ディレクトリの基本アーキテクチャ
- ITU - T勧告 X . 5 0 1 (1997) | ISO / IEC 9 5 9 4 - 2 : 1997、
Information technology - Open Systems Interconnection -The Directory: The Models .
- ITU - T勧告 X . 5 0 9 (1997) The Directory - Authentication Framework.
- ITU - T勧告 X . 5 1 1 (1997) | ISO / IEC 9 5 9 4 - 3 : 1997、
Information technology - Open Systems Interconnection -The Directory: Abstract Service Definition.
- ITU - T勧告 X . 5 1 8 (1997) | ISO / IEC 9 5 9 4 - 4 : 1997、
Information technology - Open Systems Interconnection -Distributed Procedures.
- ITU - T勧告 X . 5 1 9 (1997) | ISO / IEC 9 5 9 4 - 5 : 1997、
Information technology - Open Systems Interconnection -The Directory: Protocol Specifications.
- ITU - T勧告 X . 6 8 0 (1994) | ISO / IEC 8 8 2 4 - 1 : 1994、
Information technology - Open Systems Interconnection - Abstract Syntax Notation One (ASN.1):
Specification of basic notation.
- ITU - T勧告 X . 6 8 1 (1994) | ISO / IEC 8 8 2 4 - 2 : 1994、
Information technology - Open Systems Interconnection - Abstract Syntax Notation One (ASN.1):
Information object specification.
- ITU - T勧告 X . 6 8 2 (1994) | ISO / IEC 8 8 2 4 - 3 : 1994、
Information technology - Open Systems Interconnection - Abstract Syntax Notation One (ASN.1):
Constraint specification.
- ITU - T勧告 X . 6 8 3 (1994) | ISO / IEC 8 8 2 4 - 4 : 1994、
Information technology - Open Systems Interconnection - Abstract Syntax Notation One (ASN.1):
Parameterization of ASN.1 specifications.
- ITU - T勧告 X . 6 9 0 (1994) | ISO / IEC 8 8 2 5 - 1 : 1994、
Information technology - Open Systems Interconnection - Specification of ASN.1 encoding rules:
Basic, Canonical, and Distinguished Encoding Rules.
- ITU - T勧告 X . 8 8 0 (1994) | ISO / IEC 1 3 7 1 2 - 1 : 1994、 Information technology–
Remote Operations: Concepts, model and notation.
- ITU - T勧告 Q . 2 9 (1988) Causes of noise and ways of reducing noise in telephone exchanges.
- ITU - T勧告 Q . 7 0 0 (1993) Introduction to CCITT Signalling System No. 7.
- ITU - T勧告 Q . 7 1 0 (1988) Simplified MTP version for small systems.
- TTC標準 JT - Q 7 1 4 (1997) SCCP手順
- TTC標準 JT - Q 7 6 2 (1998) 信号と信号情報の機能概要
- TTC標準 JT - Q 7 6 3 (1998) フォーマットとコード
- ITU - T勧告 Q . 7 6 7 (1991) Application of the ISDN user part of CCITT signalling system No. 7
for international ISDN interconnections.
- TTC標準 JT - Q 7 7 1 (1997) トランザクション機能の機能内容

- T T C 標準 J T - Q 7 7 2 (1997) トランザクション機能情報要素定義
- T T C 標準 J T - Q 7 7 3 (1997) トランザクション機能のフォーマットと符合化
- T T C 標準 J T - Q 7 7 4 (1997) トランザクション機能手順
- I T U - T 勧告 Q . 7 7 5 (1993) Signalling System No. 7 - Guidelines for using Transaction Capabilities
- T T C 標準 J T - Q 7 1 1 (1997) 信号接続制御部 (S C C P) の機能
- T T C 標準 J T - Q 7 1 3 (1998) S C C P フォーマットとコード
- I T U - T 勧告 Q . 7 1 5 (1996) Signalling System No. 7 - SCCP User Guide
- T T C 標準 J T - Q 9 3 1 (1996) I S D N ユーザ・網インタフェースレイヤ 3 仕様
- T T C 標準 J T - Q 9 3 2 (1993) I S D N 付加サービス制御手順の共通原則
- I T U - T 勧告 Q . 1 2 9 0 : Intelligent Network: Glossary of terms used in the definition of Intelligent Networks.
- I T U - T 勧告 Q . 1 4 0 0 (1993) Architecture framework for the development of signalling and OA&M protocols using OSI concepts.
- I T U - T 勧告 X . 8 3 0 (1995) | I S O / I E C 1 1 5 8 6 - 1 : 1995、 Information technology – Open Systems Interconnection - Generic Upper Layers Security: Overview, Models and Notation.
- I T U - T 勧告 X . 8 3 1 (1995) | I S O / I E C 1 1 5 8 6 - 2 : 1995、 Information technology – Open Systems Interconnection - Generic Upper Layers Security: Security Exchange Service Element (SESE) Service Definition.
- I T U - T 勧告 X . 8 3 2 (1995) | I S O / I E C 1 1 5 8 6 - 3 : 1995、 Information technology – Open Systems Interconnection - Generic Upper Layers Security: Security Exchange Service Element (SESE) Protocol Specification.
- Internet Engineering Task Force (IETF) Request For Comment (RFC) 2025, "The Simple Public-Key GSS-API Mechanism (SPKM)," October 1996. <ftp://ds.internic.net/rfc/rfc2025.txt>

2.2 略語と頭字語

略語、頭字語については、本標準付属資料 B を参照のこと。

*

3. 電気通信サービスのためのインタフェース標準

3.1 総論

3.1.1 定義方法論

プロトコルの定義は、以下の3つの部分に分けられる。

- プロトコルのためのSACF/MACF規則の定義
- エンティティ間で転送されるオペレーションの定義
- 各エンティティで行われる動作の定義

SACF/MACF規則は文章記述で定義される。オペレーション定義は、抽象構文記法1(ASN.1、ITU-T勧告X.680参照)で行われており、動作は、状態遷移図で定義される。オペレーションの受信において実行される動作におけるそれ以上の手引きは、TTC標準JT-Q1224の関連する情報フローの記述から得られる。

INAPは、ROSEユーザプロトコル(ITU-T勧告X.219および229参照)である。ROSEプロトコルは、TC(TTC標準JT-Q771から774およびITU-T勧告Q.775参照)とDSS1(TTC標準JT-Q932)のコンポーネントサブレイヤに含まれる。現在、ROSE APDU(アプリケーションプロトコルデータユニット)は、No.7共通線信号方式のトランザクションサブレイヤメッセージとTTC標準JT-Q931のREGISTER、FACILITY及びDSS1の呼制御メッセージで伝達される。これをサポートする他のプロトコルは、後で追加される可能性がある。

INAP(1つのROSEユーザとして)とROSEプロトコルは、ASN.1(ITU-T勧告X.680参照)を使って規定されている。PDUへの符号化は、基本符号化規則(ITU-T勧告X.690参照)を用いるべきである。

3.1.2 物理シナリオ例

プロトコルは、機能エンティティの物理エンティティ群(PE)への任意のマッピングをサポートする。どのようにFEを可能なもっとも有利な条件で同じ場所に配置するかは決定は、ネットワークオペレータや装置製造者に任されている。なぜならば、その条件は製造者及びネットワークオペレータで変化するかもしれないからである。そのため、プロトコルは、FEがPEに対して最大限に分散することを想定(すなわち、FEにつき1つのPE)して、定義されている。

本節内で示される図は、No.7共通線信号網環境で、INAPがどのようにサポートされるかを示している。これは、INAPをサポートするために、No.7共通線信号方式だけがネットワークプロトコルとして使われるということの意味しているわけではない。

遠方に配置されたSCFとSDFの間のインタフェースは、コネクションレスSCCPとMTP(図3-1/JT-Q1228参照)のサービスを順次用いるTCを使用するINAPになるであろう。SDFは、他の種別のネットワークをアクセスするための他のプロトコルとの相互作用に対して責任がある。

TCが以降の図に現われる場合、それは(TCエンティティに対してではなく)単一のダイアログとトランザクションに関連づけられたTC機能を表しているものと解釈しなければならない。

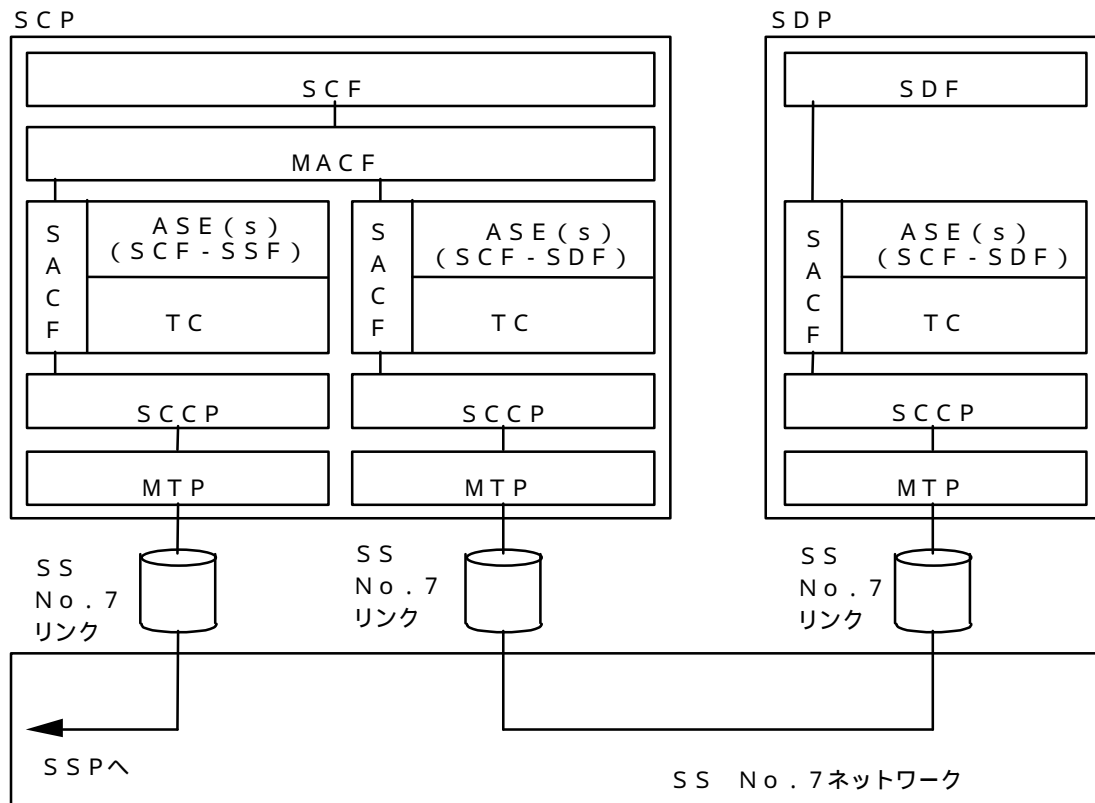


図3 - 1 / J T - Q 1 2 2 8 SCPとSDPとの間の物理的なインタフェース
(I T U - T Q . 1 2 2 8)

もし、INAPメッセージの分割・再組立が、メッセージ長のためにSCF - SDFインタフェース、SDF - SDFインタフェースに要求されるならば、TTC標準JT - Q 7 1 4にて規定されているSCCPコネクションレスメッセージの分割・再組立手順が使用されるべきである。

3.1.2.1 「SCF - 外部SRF」中継による通信の例

#

3.1.3 INAPプロトコルアーキテクチャ

この章で使われる用語の多くは、ISO IS - 9 5 4 5で定義されるOSIアプリケーションレイヤ構造に基づいている。

INAPプロトコルアーキテクチャは、図3 - 9 / J T - Q 1 2 2 8で示すように描かれ得る。

1つの物理エンティティは、他の物理エンティティと1つの相互作用をもつ(ケースa)かあるいは、複数の協調された相互作用をもつ(ケースb)。

ケースaにおいて、SACFは、使用しているASE間の協調機能を提供する。それは(受け取ったプリミティブの順序を基に)ASEによってサポートされるオペレーションの順序づけを含む。SAOは、SACFと1組のASEで表現され、1対のPEの間の単一の相互作用上で用いられる。

ケースbにおいて、MACFは、遠隔のPEにある1つのSAOと相互作用をする複数のSAOの間の協調機能を提供する。

各ASEは、1つあるいはそれ以上のオペレーションをサポートする。各オペレーションの記述は、関係するFEモデル化の動作(TTC標準JT - Q 1 2 2 4および本標準の11節から18節を参照)に結びつけられている。各オペレーションは、図3 - 1 0 / Q . 1 2 2 8で記述されるOPERATIONマクロを用いて規定される。

[T T C 標準 J T - Q 7 7 X シリーズ (トランザクション機能応用部) で定義されたように] アプリケーションコンテキストネゴシエーションメカニズムの使用は、通信している 2 つのエンティティが、それぞれの能力が何であるのか、またインタフェース上で必要とされる能力は何であるべきかを正確に識別することを可能とする。これは、インテリジェントネットワーク能力セットを通じた発展を許容するために使用されるべきである。

特定のアプリケーションコンテキストの表示が、通信している 1 対の F E によってサポートされなければ、前もってコンテキストを定めておくための何らかのメカニズムが、サポートされなければならない。

3.1.3.1 No. 7 信号システムに対する I N A P シグナリング輻輳制御

I S D N ユーザパートシグナリングで輻輳制御に対して定義されているものと同一種類の手順を適用しなければならない。シグナリング輻輳制御に対する I N A P 手順は (I T U - T 勧告 Q . 7 6 7 の D . 2 . 1 1 節) 規定されている I S D N ユーザパートシグナリング輻輳制御手順と可能な限りの整合がとられる、すなわち S C C P からの「輻輳信号局」の情報を含む N - P C S T A T E 表示プリミティブの受信時には影響のある方向へのトラヒック負荷を数段階に低減する。

上記の手順は影響のある方向へのトラヒックが M T P 信号局コードアドレッシングを採用している時のみ適用されるかもしれない。

3.1.4 I N A P アドレッシング

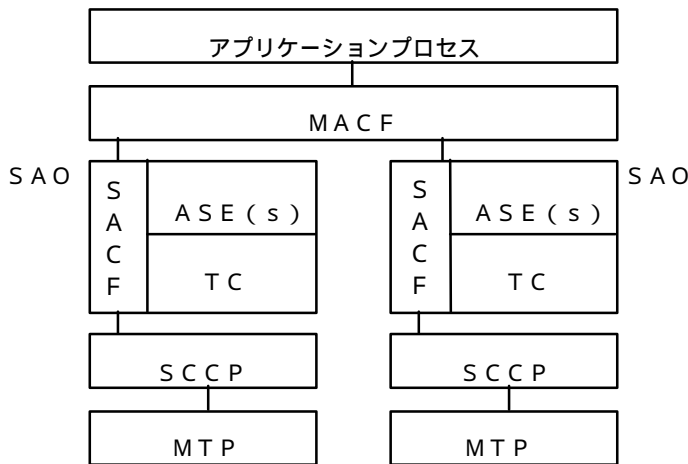
S C C P グローバルタイトルと M T P 信号局コードアドレッシング [T T C 標準 J T - Q 7 1 0 シリーズ (S C C P) および T T C 標準 J T - Q 7 0 0 シリーズ (M T P) 参照] は、それが、どの網にいるにもかかわらず、P D U がそれらの物理的な宛先 (すなわち、正しい信号局コード) に到達することを保証する。

I N A P に対して現在 1 つの S S N が T T C 標準 J T - Q 7 1 0 シリーズで割り当てられている。

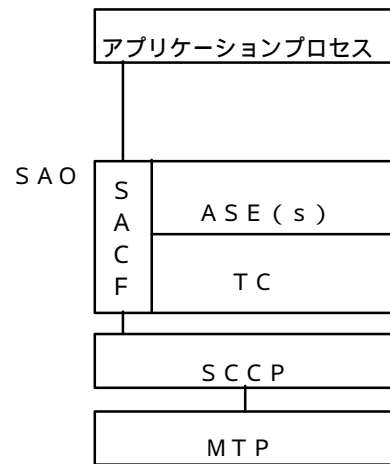
*

上記にかかわらず、S C C P によってサポートされるアドレッシング機構が使用されるであろう。

複数の調整された相互動作



単一相互動作



- SACF : 単一アソシエーション制御機能
(Single association control function)
- MACF : 複数アソシエーション制御機能
(Multiple association control function)
- SAO : 単一アソシエーションオブジェクト
(Single association object)
- ASE : アプリケーションサービス要素
(Application service element)
- INAP : インテリジェントネットワークアプリケーションプロトコル
(Intelligent network application protocol)

注 - INAPはASEsに含まれる全ての仕様の集合である。

図3 - 9 / JT - Q1228 INAPプロトコルアーキテクチャ
(ITU - T Q.1228)

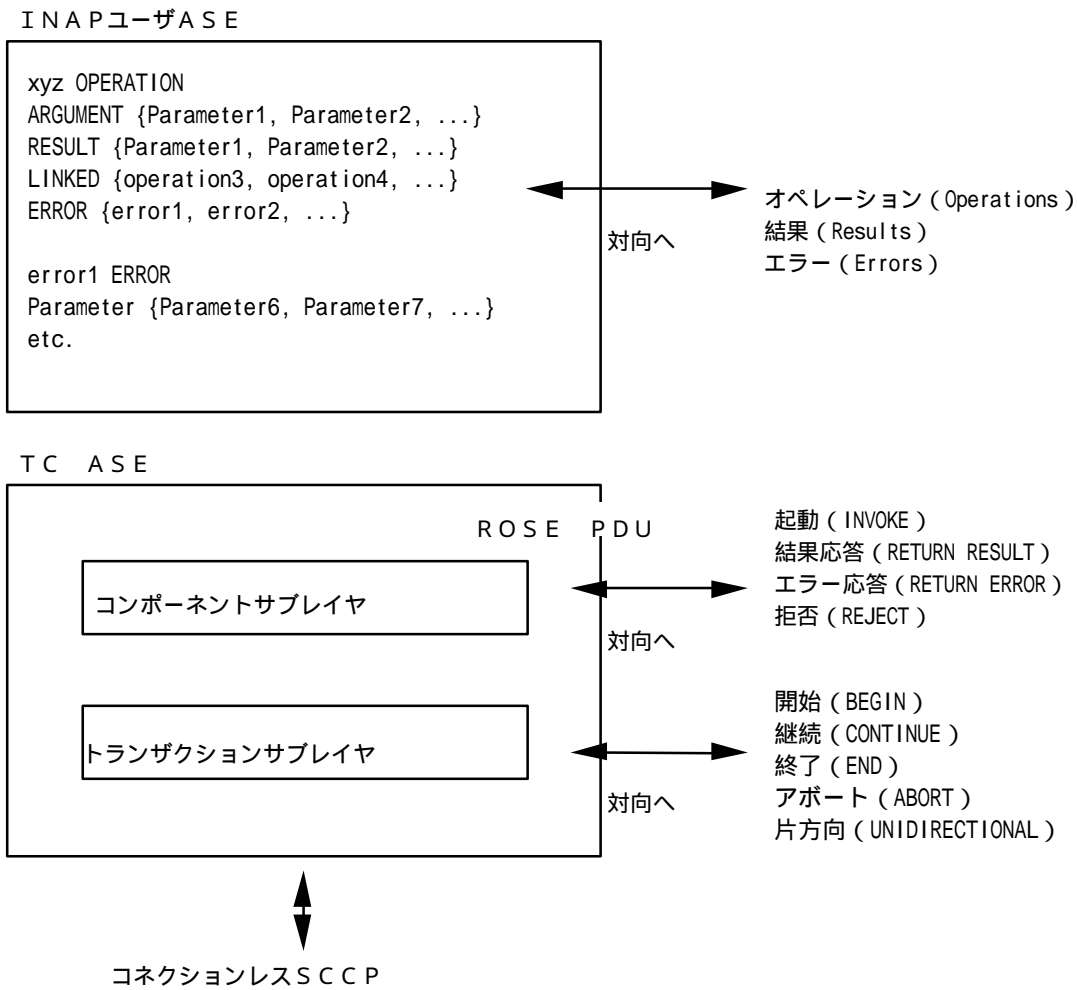


図3 - 10 / JT - Q 1 2 2 8 オペレーション記述
 (ITU - T Q . 1 2 2 8)

3.1.5 TTC標準JT - Q 1 2 2 4と本標準との関係

以下は、全ての情報フローのリストである。これらは、明記されているものを除いて、オペレーションと1対1に対応する。

オペレーションのTCダイアログやコンポーネント部へのマッピングの決定に関しては、18.1節(TCから想定されるサービス)を参照のこと。

TTC標準JT - Q 1 2 2 4参照	情報フロー	オペレーション
SCF - SDF		
12.8.2.1	Add Entry	同一
12.8.2.2	Add Entry Referral	AddEntryからのエラー応答
12.2.8.3	Add Entry Result	AddEntryからの結果応答
12.8.2.4	Authenticate	Bind
12.8.2.5	Authenticate Result	Bindからの結果応答

12.8.2.6	End Authenticated Relationship	Unbind
12.8.2.7	Execute	同一
12.8.2.8	Execute Referral	Execute からのエラー応答
12.8.2.9	Execute Result	Execute からの結果応答
12.8.2.10	Modify Entry	同一
12.8.2.11	Modify Entry Referral	ModifyEntry からのエラー応答
12.8.2.12	Modify Entry Result	ModifyEntry からの結果応答
12.8.2.13	Remove Entry	同一
12.8.2.14	Remove Entry Referral	RemoveEntry からのエラー応答
12.8.2.15	Remove Entry Result	RemoveEntry からの結果応答
12.8.2.16	Search	同一
12.8.2.17	Search Referral	Search からのエラー応答
12.8.2.18	Search Result	Search からの結果応答
SDF - SDF		
12.9.2.1	Authenticate	dSASDF Bind, dSAShadowBind
12.9.2.2	Authenticate Result	dSASDF Bind または dSAShadowBind からの結果応答
12.9.2.3	Chaining Request	chained (オペレーション)
12.9.2.4	Chaining Result	オペレーションからの結果応答
12.9.2.5	Copy Request	Coordinate Shadow Update Request Shadow Update
12.9.2.6	Copy Result	Coordinate Shadow Update または Request Shadow Update からの結果応答
12.9.2.7	End Authenticated Relationship	in-DSASDF Unbind, in-DSAShadowUnbind
12.9.2.8	Update Copy	Update Shadow
12.9.2.9	Update Copy Result	Update ShadowCopy からの結果応答

3.1.6 I N A P のために利用されるコンパチビリティメカニズム

3.1.6.1 序論

本節では、I N A P の将来バージョンとの整合を保証するために使用されるコンパチビリティメカニズムについて規定する。

以下の3つのカテゴリーのコンパチビリティが存在する。

- 将来の標準バージョンでの I N A P のマイナーな変更：
マイナーな変更は、要求された I N サービスに対して重要でない機能の変更として定義され得る。既存機能の修正の場合、呼ばれた機能が旧のものあるいは修正された別物のいずれかで実行されることが必要である。また、変更が純粋に追加である場合、それが全く実行されず、対向するアプリケーションエンティティ (A E) がその変更についての影響を知る必要のないことが必要である。マイナーな変更に対しては、新たな A C は必要とされない。
- 将来の標準バージョンでの I N A P のメジャーな変更：

メジャーな変更は、要求された I N サービスに対して重要な機能の変更として定義され得る。既存機能の修正の場合、両アプリケーションエンティティは変更となった機能の変更に関する知識をもたなければならない。変更が純粋に追加である場合、アプリケーションエンティティの 1 つでも追加された機能をサポートしていなければ、要求された I N サービスは提供されない。メジャーな変更に対しては、新たな A C が必要とされる。

- I N A P への網特有の変更：

このような追加はサービスに対してメジャーあるいはマイナーのいずれかであろう。新たな A C はこの種の変更に対して定義されないことが期待される。定義に際しては、追加は、標準の将来バージョンと同一の形式で含まれることが期待される。

3.1.6.2 I N A P コンパチビリティメカニズムの定義

3.1.6.2.1 I N A P へのメジャー追加手順

メジャー機能変更の導入をサポートするためにプロトコルは、実行される機能に関する 2 つのアプリケーション間の同調を可能とする。この同調は複雑化したフォールバック手順を避けるためどちらかのアプリケーションエンティティで新たな機能が起動される前に行われる。そのような同期を達成するために選択した解は、T T C 標準 J T - Q 7 7 3 で与えられている A C ネゴシエーション手順を用いることである。

3.1.6.2.2 I N A P へのマイナー追加手順

拡張メカニズムマーカは、将来の I N A P への標準化されたマイナー追加に用いる。このメカニズムは、タイプ定義に「拡張マーカ」(extensions marker) を含むことにより異なる拡張を実施する。この拡張はマーカの後に置かれたオプションフィールドにより表現される。エンティティがマーカの後に存在する認識できないパラメータを受け取ったとき、それらは、無視される (I T U - T 勧告 X . 6 8 x 参照) 。

3.1.6.2.3 I N A P への網特有追加の包括手順

このメカニズムは、タイプ定義の最も外側のレベルで A S N . 1 のマクロファシリティによるタイプのフィールドを明示的に宣言する能力に基づく。タイプ定義の最後に置かれた「拡張フィールド」(ExtensionField) を定義することによって動作する。この拡張フィールドは、拡張セットとして定義され、一つの拡張は任意のタイプを含むことができる。各拡張は、定義された値に関連づけられ、各拡張は着信ノードが認識しない場合そのフィールドを無視すべきかメッセージを捨てるかを定義し、これは前の節で記述された要求メカニズムの理解と同様である。このメカニズムの定義は、I T U - T 勧告 Q . 1 4 0 0 を参照のこと。

3.2 S A C F / M A C F 規則

3.2.1 T C A C の反映

T C アプリケーションコンテキストネゴシエーション規則は、提案された A C が、受け付けられれば、最初の逆方向メッセージに反映されるということを要求する。

もしその A C が受け付けられなくて、T C ユーザがダイアログを続けようとしなければ、新しいダイアログを始めるのに使うことができる代わりに A C が起動側に提供されるかもしれない。

T C A C ネゴシエーションは、S C F インタフェースにのみ適用される。

T C A C ネゴシエーションメカニズムのより詳細な記述については、J T - Q 7 7 x (トランザクション機能応用部) を参照すること。

3.2.2 オペレーションの直列 / 並列実行

いくつかの場合、オペレーションの実行が、直列か並列 (同期) かを見分ける必要がある。

同期実行ができるオペレーションは：

- 課金オペレーションは、他のいくつかのオペレーションとともに同期実行可能である。

同期されたオペレーションが示すメソッドは、いくつかのメッセージに含まれる。

上記の識別されたオペレーションの一つは、他のオペレーションがある程度まで、あるいは最後まで実行されるまで、実行してはならない。送出されて P E（通常は S C P）は、2つの別々のメッセージを送出されるオペレーションにより、制御可能である。

このメソッドは、同じメッセージを送られた全てのオペレーションが、同時に実行されなければならないこと意味するものではなく、単に同期しなければならないオペレーション（上記の識別された状態）を実行することと理解する。

上記の一般規則と F E 特有の規則の間の矛盾がある場合、3節の記述として、F E 特有の規則は一般規則を越えて適用する。

4 . 共通 I N C S - 2 型

- 4.1 データ型 #
- 4.2 エラー型 #
- 4.3 オペレーションコード #
- 4.4 エラーコード #

4.5 クラス

IN-CS2-classes {ccitt recommendation q 1228 modules(0) in-cs2-classes (4) version1(0)}

DEFINITIONS ::=

BEGIN

IMPORTS

ROS-OBJECT-CLASS, CONTRACT, OPERATION-PACKAGE, Code, OPERATION,
CONNECTION-PACKAGE

FROM Remote-Operations-Information-Objects ros-InformationObjects
emptyBind, emptyUnbind

FROM Remote-Operations-Useful-Definitions ros-UsefulDefinitions

id-package-emptyConnection,

id-rosObject-scf,

id-rosObject-cusf,

id-rosObject-dssp,

id-rosObject-srf,

id-rosObject-ssf,

ros-InformationObjects,

ros-UsefulDefinitions,

ssf-scf-Protocol,

scf-cusf-Protocol,

scf-scf-Protocol,

scf-srf-Protocol,

scf-sdf-Protocol,

datatypes

#

#

#

#

#

#

#

#

FROM IN-CS2-object-identifiers {ccitt recommendation q 1228 modules(0) in-cs2-object-identifiers (17)
version1(0)}

inCs2AssistHandoffSsfToScf,

inCs2ScfToSsfDpSpecific,

inCs2ScfToSsfGeneric,

inCs2ScfToSsfStatusReporting,

inCs2ScfToSsfTrafficManagement,

inCs2SsfToScfDpSpecific,

inCs2SsfToScfGeneric,

inCs2SsfToScfServiceManagement

FROM IN-CS2-SSF-SCF-pkgs-contracts-acscf-Protocol

cusf-scf-contract,

scf-cusf-contract

FROM IN-CS2-SCF-CUSF-pkgs-contracts-acscf-cusf-Protocol

dsspContract,

scf-scfContract

FROM IN-CS2-SCF-SCF-pkgs-contracts-acscf-scf-Protocol

srf-scf-contract

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

#

```

FROM IN-CS2-SCF-SRF-pkgs-contracts-acscf-srf-Protocol #
    dapContract
    FROM IN-CS2-SCF-SDF-Protocol scf-sdf-Protocol
    CriticalityType
FROM IN-CS2-datatypes datatypes
;

ssf ROS-OBJECT-CLASS ::= { #
    INITIATES {inCs2SsfToScfGeneric| #
                inCs2SsfToScfDpSpecific| #
                inCs2AssistHandoffSsfToScf| #
                inCs2SsfToScfServiceManagement} #
    RESPONDS {inCs2ScfToSsfGeneric| #
                inCs2ScfToSsfDpSpecific| #
                inCs2ScfToSsfTrafficManagement| #
                inCs2SsfToScfServiceManagement| #
                inCs2ScfToSsfStatusReporting} #
    ID id-rosObject-ssf} #

srf ROS-OBJECT-CLASS ::= { #
    INITIATES {srf-scf-contract} #
    ID id-rosObject-srf #
} #

cusf ROS-OBJECT-CLASS ::= { #
    INITIATES {cusf-scf-contract} #
    RESPONDS {scf-cusf-contract} #
    ID id-rosObject-cusf} #

dssp ROS-OBJECT-CLASS ::= { #
    BOTH {dsspContract} #
    ID id-rosObject-dssp #
} #

scf ROS-OBJECT-CLASS ::= {
    INITIATES {
        inCs2ScfToSsfGeneric| #
        inCs2ScfToSsfDpSpecific| #
        inCs2ScfToSsfTrafficManagement| #
        inCs2ScfToSsfServiceManagement| #
        inCs2ScfToSsfTriggerManagement| #
        inCs2ScfToSsfStatusReporting | #
-- scf to cusf contracts #
        scf-cusf-contract | #
-- scf to scf contracts #
        scf-scfContract | #
        dsspContract | #
-- sdf to scf contracts #
        dapContract #
    } #
    RESPONDS {
        inCs2SsfToScfGeneric| #
        inCs2SsfToScfDpSpecific| #
        inCs2AssistHandoffSsfToScf| #
        inCs2SsfToScfServiceManagement| #
-- cusf to scf contracts #
        cusf-scf-contract | #
-- srf to scf contracts #
        srf-scf-contract | #
-- scf to scf contracts #
        scf-scfContract | #
        dsspContract #
    } #
} #

```

```

        ID          }
                id-rosObject-scf}

EXTENSION ::= CLASS {
    &ExtensionType,
    &criticality    CriticalityType DEFAULT ignore,
    &id            Code
}
WITH SYNTAX {
    EXTENSION-SYNTAX    &ExtensionType
    CRITICALITY        &criticality
    IDENTIFIED BY      &id
}

--BOOLEANタイプで、criticality をアボート、拡張番号 1 と識別され、 ‘ Some Network Specific
--Indicator ’ という名前の拡張の追加例である。
--上記の情報をを用いたオブジェクトクラスの定義例である：
--
-- SomeNetworkSpecificIndicator    EXTENSION ::= {
--     EXTENSION-SYNTAX            BOOLEAN
--     CRITICALITY                  abort
--     IDENTIFIED BY                local : 1
-- }

-- 4 . 1 項で詳細に示した、 ExtensionField datatype を用いた転送構文の例である。
--拡張の値を TRUE 、拡張パラメータのSequence of type INTEGERを1、criticality
--ENUMERATEDを 1、そして EXPLICIT BOOLEANを TRUEに設定すると想定している。
--
-- I T U - T 勧告 Q.1 4 0 0 によって定義された拡張の使用については、今後の課題である。
--更に、拡張メカニズムマーカーは、 I N A P への将来の小さな拡張を識別するために用いられ
--る。

firstExtension EXTENSION ::= {
    EXTENSION-SYNTAX    NULL
    CRITICALITY        ignore
    IDENTIFIED BY      local:1
}
-- firstExtension is just an example.
SupportedExtensions {PARAMETERS-BOUND : bound} EXTENSION ::= {firstExtension , ...
--ネットワークオペレータ拡張のフルセット –
}
-- SupportedExtension is the full set of the network operator extensions.

UISCRIPT ::= CLASS {
    &SpecificInfo          OPTIONAL,
    &Result                OPTIONAL,
    &id                    Code
}

WITH SYNTAX {
    [WITH-SPECIFICINFO    &SpecificInfo]
    [WITH-RESULT          &Result]
    IDENTIFIED BY        &id
}

firstScript UISCRIPT ::=
{
    IDENTIFIED BY local:1
}
-- firstScript is just an example.

```

```

SupportedUIScripts {PARAMETERS-BOUND : bound}  UISCRIPT ::= {firstScript , ...      #
-- ユーザ相互動作スクリプトのフルセット—                                         #
}                                                                                       #
-- SupportedUIScripts はユーザ相互動作スクリプトのフルセット                         #
inEmptyUnbind OPERATION ::= {
    RETURN RESULT FALSE
    ALWAYS RESPONDS FALSE}
emptyConnectionPackage CONNECTION-PACKAGE ::= {
    BIND emptyBind
    UNBIND inEmptyUnbind
    RESPONDER UNBIND TRUE
    ID id-package-emptyConnection
}
PARAMETERS-BOUND ::= CLASS                                                           #
{                                                                                       #
    &minAChBillingChargingLength INTEGER,                                           #
    &maxAChBillingChargingLength INTEGER,                                           #
    &minAttributesLength INTEGER,                                                  #
    &maxAttributesLength INTEGER,                                                  #
    &minBackwardGVNSLength INTEGER,                                                #
    &maxBackwardGVNSLength INTEGER,                                                #
    &maxBearerCapabilityLength INTEGER,                                            #
    &minCalledPartyNumberLength INTEGER,                                           #
    &maxCalledPartyNumberLength INTEGER,                                           #
    &minCallingPartyNumberLength INTEGER,                                          #
    &maxCallingPartyNumberLength INTEGER,                                          #
    &minCallResultLength INTEGER,                                                  #
    &maxCallResultLength INTEGER,                                                  #
    &maxCauseLength INTEGER,                                                       #
    &minDigitsLength INTEGER,                                                      #
    &maxDigitsLength INTEGER,                                                      #
    &minDisplayInformationLength INTEGER,                                          #
    &maxDisplayInformationLength INTEGER,                                          #
    &minEventSpecificInformationChargingLength INTEGER,                           #
    &maxEventSpecificInformationChargingLength INTEGER,                           #
    &minEventTypeChargingLength INTEGER,                                           #
    &maxEventTypeChargingLength INTEGER,                                           #
    &minFCIBillingChargingLength INTEGER,                                          #
    &maxFCIBillingChargingLength INTEGER,                                          #
    &minForwardGVNSLength INTEGER,                                                 #
    &maxForwardGVNSLength INTEGER,                                                 #
    &minGenericNameLength INTEGER,                                                 #
    &maxGenericNameLength INTEGER,                                                 #
    &minGenericNumberLength INTEGER,                                               #
    &maxGenericNumberLength INTEGER,                                               #
    &maxInitialTimeInterval INTEGER,                                               #
    &maxINServiceCompatibilityIndLength INTEGER,                                  #
    &minIPAvailableLength INTEGER,                                                 #
    &maxIPAvailableLength INTEGER,                                                 #
    &minIPSSPCapabilitiesLength INTEGER,                                           #
    &maxIPSSPCapabilitiesLength INTEGER,                                           #
    &minLocationNumberLength INTEGER,                                              #
    &maxLocationNumberLength INTEGER,                                              #
    &minMailBoxIDLength INTEGER,                                                  #
    &maxMailBoxIDLength INTEGER,                                                  #
    &minMessageContentLength INTEGER,                                             #
    &maxMessageContentLength INTEGER,                                             #
    &minMidCallControlInfoNum INTEGER,                                            #
    &maxMidCallControlInfoNum INTEGER,                                            #
    &minOriginalCalledPartyIDLength INTEGER,                                       #
    &maxOriginalCalledPartyIDLength INTEGER,                                       #
    &minReasonLength INTEGER,                                                      #

```


&maxReasonLength	INTEGER,	#
&minReceivedInformationLength	INTEGER,	#
&maxReceivedInformationLength	INTEGER,	#
&maxRecordedMessageUnits	INTEGER,	#
&maxRecordingTime	INTEGER,	#
&minRedirectingPartyIDLength	INTEGER,	#
&maxRedirectingPartyIDLength	INTEGER,	#
&minRequestedUTSINum	INTEGER,	#
&maxRequestedUTSINum	INTEGER,	#
&minRouteListLength	INTEGER,	#
&maxRouteListLength	INTEGER,	#
&minScfIDLength	INTEGER,	#
&maxScfIDLength	INTEGER,	#
&minScfAddressLength	INTEGER,	#
&maxScfAddressLength	INTEGER,	#
&minSCIBillingChargingLength	INTEGER,	#
&maxSCIBillingChargingLength	INTEGER,	#
&minServiceInteractionIndicatorsLength	INTEGER,	#
&maxServiceInteractionIndicatorsLength	INTEGER,	#
&minSFBillingChargingLength	INTEGER,	#
&maxSFBillingChargingLength	INTEGER,	#
&minUSIInformationLength	INTEGER,	#
&maxUSIInformationLength	INTEGER,	#
&minUSIServiceIndicatorLength	INTEGER,	#
&maxUSIServiceIndicatorLength	INTEGER,	#
&numOfBCSMEvents	INTEGER,	#
&numOfBCUSMEvents	INTEGER,	#
&numOfChargingEvents	INTEGER,	#
&numOfCSAs	INTEGER,	#
&numOfCSs	INTEGER,	#
&numOfExtensions	INTEGER,	#
&numOfGenericNumbers	INTEGER,	#
&numOfInServiceCompatibilityIndLength	INTEGER,	#
&numOfLegs	INTEGER,	#
&numOfMessageIDs	INTEGER,	#
&maxAmount	INTEGER,	#
&maxInitialUnitIncrement	INTEGER,	#
&maxScalingFactor	INTEGER,	#
&maxSegmentsPerDataInterval	INTEGER,	#
&maxTimePerInterval	INTEGER,	#
&maxUnitsPerDataInterval	INTEGER,	#
&maxUnitsPerInterval	INTEGER,	#
&ub-maxUserCredit	INTEGER,	#
&ub-nbCall	INTEGER	#
}		#
WITH SYNTAX		#
{		#
MINIMUM-FOR-ACH-BILLING-CHARGING	&minAChBillingChargingLength	#
MAXIMUM-FOR-ACH-BILLING-CHARGING	&maxAChBillingChargingLength	#
MINIMUM-FOR-ATTRIBUTES	&minAttributesLength	#
MAXIMUM-FOR-ATTRIBUTES	&maxAttributesLength	#
MAXIMUM-FOR-BACKWARD-GVNS	&minBackwardGVNSLength	#
MAXIMUM-FOR-BACKWARD-GVNS	&maxBackwardGVNSLength	#
MAXIMUM-FOR-BEARER-CAPABILITY	&maxBearerCapabilityLength	#
MINIMUM-FOR-CALLED-PARTY-NUMBER	&minCalledPartyNumberLength	#
MAXIMUM-FOR-CALLED-PARTY-NUMBER	&maxCalledPartyNumberLength	#
MINIMUM-FOR-CALLING-PARTY-NUMBER	&minCallingPartyNumberLength	#
MAXIMUM-FOR-CALLING-PARTY-NUMBER	&maxCallingPartyNumberLength	#
MINIMUM-FOR-CALL-RESULT	&minCallResultLength	#
MAXIMUM-FOR-CALL-RESULT	&maxCallResultLength	#
MAXIMUM-FOR-CAUSE	&maxCauseLength	#
MINIMUM-FOR-DIGITS	&minDigitsLength	#
MAXIMUM-FOR-DIGITS	&maxDigitsLength	#

MINIMUM-FOR-DISPLAY	&minDisplayInformationLength	#
MAXIMUM-FOR-DISPLAY	&maxDisplayInformationLength	#
MINIMUM-FOR-EVENT-SPECIFIC-CHARGING		#
	&minEventSpecificInformationChargingLength	#
MAXIMUM-FOR-EVENT-SPECIFIC-CHARGING		#
	&maxEventSpecificInformationChargingLength	#
MINIMUM-FOR-EVENT-TYPE-CHARGING	&minEventTypeChargingLength	#
MAXIMUM-FOR-EVENT-TYPE-CHARGING	&maxEventTypeChargingLength	#
MINIMUM-FOR-FCI-BILLING-CHARGING	&minFCIBillingChargingLength	#
MAXIMUM-FOR-FCI-BILLING-CHARGING	&maxFCIBillingChargingLength	#
MINIMUM-FOR-FORWARD-GVNS	&minForwardGVNSLength	#
MAXIMUM-FOR-FORWARD-GVNS	&maxForwardGVNSLength	#
MINIMUM-FOR-GENERIC-NAME	&minGenericNameLength	#
MAXIMUM-FOR-GENERIC-NAME	&maxGenericNameLength	#
MINIMUM-FOR-GENERIC-NUMBER	&minGenericNumberLength	#
MAXIMUM-FOR-GENERIC-NUMBER	&maxGenericNumberLength	#
MAXIMUM-FOR-INITIAL-TIME-INTERVAL	&maxInitialTimeInterval	#
MAXIMUM-FOR-IN-SERVICE-COMPATIBILITY	&maxINSserviceCompatibilityIndLength	#
MINIMUM-FOR-IP-AVAILABLE	&minIPAvailableLength	#
MAXIMUM-FOR-IP-AVAILABLE	&maxIPAvailableLength	#
MINIMUM-FOR-IP-SSP-CAPABILITIES	&minIPSSPCapabilitiesLength	#
MAXIMUM-FOR-IP-SSP-CAPABILITIES	&maxIPSSPCapabilitiesLength	#
MINIMUM-FOR-LOCATION-NUMBER	&minLocationNumberLength	#
MAXIMUM-FOR-LOCATION-NUMBER	&maxLocationNumberLength	#
MINIMUM-FOR-MAIL-BOX-ID	&minMailBoxIDLength	#
MAXIMUM-FOR-MAIL-BOX-ID	&maxMailBoxIDLength	#
MINIMUM-FOR-MESSAGE-CONTENT	&minMessageContentLength	#
MAXIMUM-FOR-MESSAGE-CONTENT	&maxMessageContentLength	#
MINIMUM-FOR-MID-CALL-CONTROL-INFO	&minMidCallControlInfoNum	#
MAXIMUM-FOR-MID-CALL-CONTROL-INFO	&maxMidCallControlInfoNum	#
MINIMUM-FOR-ORIGINAL-CALLED-PARTY-ID	&minOriginalCalledPartyIDLength	#
MAXIMUM-FOR-ORIGINAL-CALLED-PARTY-ID	&maxOriginalCalledPartyIDLength	#
MINIMUM-FOR-REASON	&minReasonLength	#
MAXIMUM-FOR-REASON	&maxReasonLength	#
MINIMUM-FOR-RECEIVED-INFORMATION	&minReceivedInformationLength	#
MAXIMUM-FOR-RECEIVED-INFORMATION	&maxReceivedInformationLength	#
MAXIMUM-FOR-RECORDED-MESSAGE-UNITS	&maxRecordedMessageUnits	#
MAXIMUM-FOR-RECORDING-TIME	&maxRecordingTime	#
MINIMUM-FOR-REDIRECTING-ID	&minRedirectingPartyIDLength	#
MAXIMUM-FOR-REDIRECTING-ID	&maxRedirectingPartyIDLength	#
MINIMUM-FOR-REQUESTED-UTSI-NUM	&minRequestedUTSIUM	#
MAXIMUM-FOR-REQUESTED-UTSI-NUM	&maxRequestedUTSIUM	#
MINIMUM-FOR-ROUTE-LIST	&minRouteListLength	#
MAXIMUM-FOR-ROUTE-LIST	&maxRouteListLength	#
MINIMUM-FOR-SCF-ID	&minScfIDLength	#
MAXIMUM-FOR-SCF-ID	&maxScfIDLength	#
MINIMUM-FOR-SCF-ADDRESS	&minScfAddressLength	#
MAXIMUM-FOR-SCF-ADDRESS	&maxScfAddressLength	#
MINIMUM-FOR-SCI-BILLING-CHARGING	&minSCIBillingChargingLength	#
MAXIMUM-FOR-SCI-BILLING-CHARGING	&maxSCIBillingChargingLength	#
MINIMUM-FOR-SII	&minServiceInteractionIndicatorsLength	#
MAXIMUM-FOR-SII	&maxServiceInteractionIndicatorsLength	#
MINIMUM-FOR-SF-BILLING-CHARGING	&minSFBillingChargingLength	#
MAXIMUM-FOR-SF-BILLING-CHARGING	&maxSFBillingChargingLength	#
MINIMUM-FOR-USI-INFORMATION	&minUSIInformationLength	#
MAXIMUM-FOR-USI-INFORMATION	&maxUSIInformationLength	#
MINIMUM-FOR-USI-SERVICE-INDICATOR	&minUSIServiceIndicatorLength	#
MAXIMUM-FOR-USI-SERVICE-INDICATOR	&maxUSIServiceIndicatorLength	#
NUM-OF-BCSM-EVENT	&numOfBCSMEvents	#
NUM-OF-BCUSM-EVENT	&numOfBCUSMEvents	#
NUM-OF-CHARGING-EVENT	&numOfChargingEvents	#
NUM-OF-CSAS	&numOfCSAs	#
NUM-OF-CSS	&numOfCSSs	#

```

NUM-OF-EXTENSIONS                &numOfExtensions                #
NUM-OF-GENERIC-NUMBERS           &numOfGenericNumbers           #
NUM-OF-IN-SERVICE-COMPATIBILITY-ID &numOfInServiceCompatibilityIndLength #
NUM-OF-LEGS                      &numOfLegs                     #
NUM-OF-MESSAGE-IDS              &numOfMessageIDs              #
MAXIMUM-FOR-AMOUNT              &maxAmount                     #
MAXIMUM-FOR-INITIAL-UNIT-INCREMENT &maxInitialUnitIncrement      #
MAXIMUM-FOR-SCALING-FACTOR      &maxScalingFactor             #
MAXIMUM-FOR-SEGMENTS-PER-DATA-INTERVAL &maxSegmentsPerDataInterval  #
MAXIMUM-FOR-TIME-PER-INTERVAL   &maxTimePerInterval           #
MAXIMUM-FOR-UNITS-PER-DATA-INTERVAL &maxUnitsPerDataInterval     #
MAXIMUM-FOR-UNITS-PER-INTERVAL  &maxUnitsPerInterval          #
MAXIMUM-FOR-UB-USER-CREDIT      &ub-maxUserCredit             #
MAXIMUM-FOR-UB-NB-CALL          &ub-nbCall                    #
}
}
}
-- The following instance of the parameter bound is just an example
networkSpecificBoundSet PARAMETERS-BOUND ::=
{
    MINIMUM-FOR-ACH-BILLING-CHARGING 1 -- example value #
    MAXIMUM-FOR-ACH-BILLING-CHARGING 5 -- example value #
    MINIMUM-FOR-ATTRIBUTES 1 -- example value #
    MAXIMUM-FOR-ATTRIBUTES 5 -- example value #
    MAXIMUM-FOR-BACKWARD-GVNS 1 -- example value #
    MAXIMUM-FOR-BACKWARD-GVNS 5 -- example value #
    MAXIMUM-FOR-BEARER-CAPABILITY 5 -- example value #
    MINIMUM-FOR-CALLED-PARTY-NUMBER 1 -- example value #
    MAXIMUM-FOR-CALLED-PARTY-NUMBER 5 -- example value #
    MINIMUM-FOR-CALLING-PARTY-NUMBER 1 -- example value #
    MAXIMUM-FOR-CALLING-PARTY-NUMBER 5 -- example value #
    MINIMUM-FOR-CALL-RESULT 1 -- example value #
    MAXIMUM-FOR-CALL-RESULT 5 -- example value #
    MAXIMUM-FOR-CAUSE 4 -- example value #
    MINIMUM-FOR-DIGITS 1 -- example value #
    MAXIMUM-FOR-DIGITS 5 -- example value #
    MINIMUM-FOR-DISPLAY 1 -- example value #
    MAXIMUM-FOR-DISPLAY 5 -- example value #
    MINIMUM-FOR-EVENT-SPECIFIC-CHARGING 1 -- example value #
    MAXIMUM-FOR-EVENT-SPECIFIC-CHARGING 5 -- example value #
    MINIMUM-FOR-EVENT-TYPE-CHARGING 1 -- example value #
    MAXIMUM-FOR-EVENT-TYPE-CHARGING 5 -- example value #
    MINIMUM-FOR-FCI-BILLING-CHARGING 1 -- example value #
    MAXIMUM-FOR-FCI-BILLING-CHARGING 5 -- example value #
    MINIMUM-FOR-FORWARD-GVNS 1 -- example value #
    MAXIMUM-FOR-FORWARD-GVNS 5 -- example value #
    MINIMUM-FOR-GENERIC-NAME 1 -- example value #
    MAXIMUM-FOR-GENERIC-NAME 5 -- example value #
    MINIMUM-FOR-GENERIC-NUMBER 1 -- example value #
    MAXIMUM-FOR-GENERIC-NUMBER 5 -- example value #
    MAXIMUM-FOR-INITIAL-TIME-INTERVAL 5 -- example value #
    MAXIMUM-FOR-IN-SERVICE-COMPATIBILITY 5 -- example value #
    MINIMUM-FOR-IP-AVAILABLE 1 -- example value #
    MAXIMUM-FOR-IP-AVAILABLE 5 -- example value #
    MINIMUM-FOR-IP-SSP-CAPABILITIES 1 -- example value #
    MAXIMUM-FOR-IP-SSP-CAPABILITIES 5 -- example value #
    MINIMUM-FOR-LOCATION-NUMBER 1 -- example value #
    MAXIMUM-FOR-LOCATION-NUMBER 5 -- example value #
    MINIMUM-FOR-MAIL-BOX-ID 1 -- example value #
    MAXIMUM-FOR-MAIL-BOX-ID 5 -- example value #
    MINIMUM-FOR-MESSAGE-CONTENT 1 -- example value #
    MAXIMUM-FOR-MESSAGE-CONTENT 5 -- example value #
    MINIMUM-FOR-MID-CALL-CONTROL-INFO 1 -- example value #
    MAXIMUM-FOR-MID-CALL-CONTROL-INFO 5 -- example value #
}
}
}

```

MINIMUM-FOR-ORIGINAL-CALLED-PARTY-ID	1	-- example value	#
MAXIMUM-FOR-ORIGINAL-CALLED-PARTY-ID	5	-- example value	#
MINIMUM-FOR-REASON	1	-- example value	#
MAXIMUM-FOR-REASON	5	-- example value	#
MINIMUM-FOR-RECEIVED-INFORMATION	1	-- example value	#
MAXIMUM-FOR-RECEIVED-INFORMATION	5	-- example value	#
MAXIMUM-FOR-RECORDED-MESSAGE-UNITS	5	-- example value	#
MAXIMUM-FOR-RECORDING-TIME	5	-- example value	#
MINIMUM-FOR-REDIRECTING-ID	1	-- example value	#
MAXIMUM-FOR-REDIRECTING-ID	5	-- example value	#
MINIMUM-FOR-REQUESTED-UTSI-NUM	1	-- example value	#
MAXIMUM-FOR-REQUESTED-UTSI-NUM	5	-- example value	#
MINIMUM-FOR-ROUTE-LIST	1	-- example value	#
MAXIMUM-FOR-ROUTE-LIST	5	-- example value	#
MINIMUM-FOR-SCF-ID	1	-- example value	#
MAXIMUM-FOR-SCF-ID	5	-- example value	#
MINIMUM-FOR-SCF-ADDRESS	1	-- example value	#
MAXIMUM-FOR-SCF-ADDRESS	5	-- example value	#
MINIMUM-FOR-SCI-BILLING-CHARGING	1	-- example value	#
MAXIMUM-FOR-SCI-BILLING-CHARGING	5	-- example value	#
MINIMUM-FOR-SII	1	-- example value	#
MAXIMUM-FOR-SII	5	-- example value	#
MINIMUM-FOR-SF-BILLING-CHARGING	1	-- example value	#
MAXIMUM-FOR-SF-BILLING-CHARGING	5	-- example value	#
MINIMUM-FOR-USI-INFORMATION	1	-- example value	#
MAXIMUM-FOR-USI-INFORMATION	5	-- example value	#
MINIMUM-FOR-USI-SERVICE-INDICATOR	1	-- example value	#
MAXIMUM-FOR-USI-SERVICE-INDICATOR	5	-- example value	#
NUM-OF-BCSM-EVENT	4	-- example value	#
NUM-OF-BCUSM-EVENT	4	-- example value	#
NUM-OF-CHARGING-EVENT	4	-- example value	#
NUM-OF-CSAS	2	-- example value	#
NUM-OF-CSS	2	-- example value	#
NUM-OF-EXTENSIONS	1	-- example value	#
NUM-OF-GENERIC-NUMBERS	2	-- example value	#
NUM-OF-IN-SERVICE-COMPATIBILITY-ID	2	-- example value	#
NUM-OF-LEGS	2	-- example value	#
NUM-OF-MESSAGE-IDS	2	-- example value	#
NUM-OF-RECORDED-MESSAGE-IDS	2	-- example value	#
MAXIMUM-FOR-AMOUNT	2	-- example value	#
MAXIMUM-FOR-INITIAL-UNIT-INCREMENT	2	-- example value	#
MAXIMUM-FOR-SCALING-FACTOR	2	-- example value	#
MAXIMUM-FOR-SEGMENTS-PER-DATA-INTERVAL	5	-- example value	#
MAXIMUM-FOR-TIME-PER-INTERVAL	5	-- example value	#
MAXIMUM-FOR-UNITS-PER-DATA-INTERVAL	5	-- example value	#
MAXIMUM-FOR-UNITS-PER-INTERVAL	5	-- example value	#
MAXIMUM-FOR-UB-USER-CREDIT	5	-- example value	#
MAXIMUM-FOR-UB-NB-CALL	5	-- example value	#

}
END

4.6 オブジェクト識別子

```
IN-CS2-object-identifiers {ccitt recommendation q 1228 modules(0) in-cs2-object-identifiers(17)
version1(0)}
DEFINITIONS ::=

BEGIN
--このモジュールは、 I N C S - 2 のモジュール、 パッケージ、 コントラクトとアプリケーション
ンコンテキストのためにオブジェクト識別子を割り当てる。

-- For Modules from TCAP, ROS,
tc-Messages                OBJECT IDENTIFIER ::=
    {ccitt recommendation q 773 modules(2) messages(1) version3(3)}
tc-NotationExtensions      OBJECT IDENTIFIER ::=
    {ccitt recommendation q 775 modules(2) notation-extension (4) version1(1)}
ros-InformationObjects     OBJECT IDENTIFIER ::=
    {joint-iso-ccitt remote-operations(4) informationObjects(5) version1(0)}
ros-genericPDUs            OBJECT IDENTIFIER ::=
    {joint-iso-ccitt remote-operations(4) generic-ROS-PDUs(6) version1(0)}
ros-UsefulDefinitions     OBJECT IDENTIFIER ::=
    {joint-iso-ccitt remote-operations(4) useful-definitions(7) version1(0)}
sese-APDUs                 OBJECT IDENTIFIER ::=                                #
    {joint-iso-ccitt genericULS(20) modules(1) seseAPDUs(6) }                #
guls-Notation              OBJECT IDENTIFIER ::=
    {joint-iso-ccitt genericULS (20) modules (1) notation (1)}
guls-SecurityTransformations OBJECT IDENTIFIER ::=                                #
    {joint-iso-itu-t genericULS (20) modules (1) gulsSecurityTransformations (3) } #
ds-UsefulDefinitions      OBJECT IDENTIFIER ::=
    {joint-iso-ccitt ds(5) module(1) usefulDefinitions(0) 3 }
spkmGssTokens              OBJECT IDENTIFIER ::=                                #
    { iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) spkm(1) #
    spkmGssTokens(10) }                                                    #

-- For IN-CS1 Modules
contexts                    OBJECT IDENTIFIER ::=
    {ccitt recommendation q 1218 modules (0) contexts (8) selectedContexts (1) version (1)}

-- For IN CS2 Modules
datatypes                  OBJECT IDENTIFIER ::=
    {ccitt recommendation q 1228 modules(0) in-cs2-datatypes (0) version1(0)}
errortypes                 OBJECT IDENTIFIER ::=
    {ccitt recommendation q 1228 modules(0) in-cs2-errortypes (1) version1(0)}
operationcodes             OBJECT IDENTIFIER ::=
    {ccitt recommendation q 1228 modules(0) in-cs2-operationcodes (2) version1(0)}
errorcodes                 OBJECT IDENTIFIER ::=
    {ccitt recommendation q 1228 modules(0) in-cs2-errorcodes (3) version1(0)}
classes                    OBJECT IDENTIFIER ::=
    {ccitt recommendation q 1228 modules(0) in-cs2-classes (4) version1(0)}
ssf-scf-Operations         OBJECT IDENTIFIER ::=                                #
    {ccitt recommendation q 1228 modules(0) in-cs2-ssf-scf-ops-args (5) version1(0)} #
ssf-scf-Protocol           OBJECT IDENTIFIER ::=                                #
    {ccitt recommendation q 1228 modules(0) in-cs2-ssf-scf-pkgs-contracts-acs (6) version1(0)} #
scf-srf-Operations        OBJECT IDENTIFIER ::=                                #
    {ccitt recommendation q 1228 modules(0) in-cs2-scf-srf-ops-args (7) version1(0)} #
scf-srf-Protocol           OBJECT IDENTIFIER ::=                                #
    {ccitt recommendation q 1228 modules(0) in-cs2-scf-srf-pkgs-contracts-acs(8) version1(0)} #
sdf-InformationFramework  OBJECT IDENTIFIER ::=
    { ccitt recommendation q 1228 module(0) sdfInformationFramework(9) version1(0) }
sdf-BasicAccessControl     OBJECT IDENTIFIER ::=
    { ccitt recommendation q 1228 module(0) sdfBasicAccessControl(10) version1(0) }
scf-sdf-Operations        OBJECT IDENTIFIER ::=
    {ccitt recommendation q 1228 module(0) scf-sdf-operations(11) version1(0) }
```

```

scf-sdf-Protocol                OBJECT IDENTIFIER ::=
    {ccitt recommendation q 1218 modules(0) in-scf-sdf-protocol(12) version1(0)}
scf-scf-Operations              OBJECT IDENTIFIER ::= #
    {ccitt recommendation q 1228 modules(0) in-cs2-scf-scf-ops-args (13) version1(0)} #
scf-scf-Protocol                OBJECT IDENTIFIER ::= #
    {ccitt recommendation q 1228 modules(0) in-cs2-scf-scf-pkgs-contracts-acs (14) version1(0)} #
scf-cusf-Operations            OBJECT IDENTIFIER ::= #
    {ccitt recommendation q 1228 modules(0) in-cs2-scf-cusf-ops-args (15) version1(0)} #
scf-cusf-Protocol              OBJECT IDENTIFIER ::= #
    {ccitt recommendation q 1228 modules(0) in-cs2-scf-cusf-pkgs-contracts-acs (16) version1(0)} #
object-identifiers             OBJECT IDENTIFIER ::=
    {ccitt recommendation q 1228 modules(0) in-cs2-object-identifiers(17) version1(0)}
sdf-sdf-Protocol                OBJECT IDENTIFIER ::=
    { ccitt recommendation q 1228 module(0) in-cs2-sdf-sdf-Protocol(18) version1(0) }

id-cs2                          OBJECT IDENTIFIER ::= {ccitt recommendation q 1228 cs2 (2)}

id-ac                           OBJECT IDENTIFIER ::= {id-cs2 ac(3)}
id-as                           OBJECT IDENTIFIER ::= {id-cs2 as(5)}
id-rosObject                    OBJECT IDENTIFIER ::= {id-cs2 rosObject(25)}
id-contract                     OBJECT IDENTIFIER ::= {id-cs2 contract(26)}
id-package                      OBJECT IDENTIFIER ::= {id-cs2 package(27)}

-- for ac, as, rosObject, contract and package, the values are identical to Q1218 #
id-package-scf-scfConnection    OBJECT IDENTIFIER ::= {id-package 46} #
id-package-dsspConnection       OBJECT IDENTIFIER ::= {id-package 47} #

-- ROS Objects
id-rosObject-scf                OBJECT IDENTIFIER ::= {id-rosObject 4}
id-rosObject-ssf                OBJECT IDENTIFIER ::= {id-rosObject 5} #
id-rosObject-srf                OBJECT IDENTIFIER ::= {id-rosObject 6} #
id-rosObject-cusf               OBJECT IDENTIFIER ::= {id-rosObject 7} #
id-rosObject-dssp               OBJECT IDENTIFIER ::= {id-rosObject 8} #
id-rosObject-sdf                OBJECT IDENTIFIER ::= {id-rosObject 9}
id-rosObject-dua                OBJECT IDENTIFIER ::= {id-rosObject 1}
id-rosObject-directory          OBJECT IDENTIFIER ::= {id-rosObject 2}
id-rosObject-dapDSA             OBJECT IDENTIFIER ::= {id-rosObject 3}
id-rosObject-dspDSA             OBJECT IDENTIFIER ::= {id-rosObject 10}
id-rosObject-initiatingConsumerDSA OBJECT IDENTIFIER ::= {id-rosObject 11}
id-rosObject-respondingSupplierDSA OBJECT IDENTIFIER ::= {id-rosObject 12}
id-rosObject-respondingConsumerDSA OBJECT IDENTIFIER ::= {id-rosObject 13}
id-rosObject-initiatingSupplierDSA OBJECT IDENTIFIER ::= {id-rosObject 14}

-- ssf/scf Application Context #
id-ac-cs2-ssf-scfGenericAC      OBJECT IDENTIFIER ::= {id-ac 4} #
id-ac-cs2-ssf-scfDPSpecificAC   OBJECT IDENTIFIER ::= {id-ac 5} #
id-ac-cs2-ssf-scfAssistHandoffAC OBJECT IDENTIFIER ::= {id-ac 6} #
id-ac-cs2-ssf-scfServiceManagementAC OBJECT IDENTIFIER ::= {id-ac 7} #
id-ac-cs2-scf-ssfGenericAC      OBJECT IDENTIFIER ::= {id-ac 8} #
id-ac-cs2-scf-ssfDPSpecificAC   OBJECT IDENTIFIER ::= {id-ac 9} #
id-ac-cs2-scf-ssfTrafficManagementAC OBJECT IDENTIFIER ::= {id-ac 10} #
id-ac-cs2-scf-ssfServiceManagementAC OBJECT IDENTIFIER ::= {id-ac 11} #
id-ac-cs2-scf-ssfStatusReportingAC OBJECT IDENTIFIER ::= {id-ac 12} #
id-ac-cs2-scf-ssfTriggerManagementAC OBJECT IDENTIFIER ::= {id-ac 13} #

-- srf/scf Application Context #
id-ac-srf-scf                   OBJECT IDENTIFIER ::= {id-ac 14} #

-- SCF-SDF - application contexts --
id-ac-indirectoryAccessAC       OBJECT IDENTIFIER ::= {id-ac 1}
id-ac-indirectoryAccessWith3seAC OBJECT IDENTIFIER ::= {id-ac 2} #
id-ac-inExtendedDirectoryAccessAC OBJECT IDENTIFIER ::= {id-ac 3}
id-ac-inExtendedDirectoryAccessWith3seAC OBJECT IDENTIFIER ::= {id-ac 27} #

```

```

--SDF - SDF Application Contexts
id-ac-indirectorySystemAC          OBJECT IDENTIFIER ::= { id-ac 15 }
id-ac-inShadowSupplierInitiatedAC  OBJECT IDENTIFIER ::= { id-ac 16 }
id-ac-inShadowConsumerInitiatedAC  OBJECT IDENTIFIER ::= { id-ac 17 }
id-ac-indirectorySystemWith3seAC    OBJECT IDENTIFIER ::= { id-ac 18 }      #
id-ac-inShadowSupplierInitiatedWith3seAC  OBJECT IDENTIFIER ::= { id-ac 19 }      #
id-ac-inShadowConsumerInitiatedWith3seAC  OBJECT IDENTIFIER ::= { id-ac 20 }      #

-- scf/scf Application Context
id-ac-scf-scfOperationsAC           OBJECT IDENTIFIER ::= { id-ac 21 }      #
id-ac-distributedSCFSystemAC        OBJECT IDENTIFIER ::= { id-ac 22 }      #
id-ac-scf-scfOperationsWith3seAC    OBJECT IDENTIFIER ::= { id-ac 23 }      #
id-ac-distributedSCFSystemWith3seAC  OBJECT IDENTIFIER ::= { id-ac 24 }      #

-- cusf/scf Application Context
id-ac-scf-cusf                      OBJECT IDENTIFIER ::= { id-ac 25 }      #
id-ac-cusf-scf                      OBJECT IDENTIFIER ::= { id-ac 26 }      #

-- ssf/scf Contracts
id-inCs2SsfToScfGeneric             OBJECT IDENTIFIER ::= { id-contract 3 } #
id-inCs2SsfToScfDpSpecific          OBJECT IDENTIFIER ::= { id-contract 4 } #
id-inCs2AssistHandoffSsfToScf      OBJECT IDENTIFIER ::= { id-contract 5 } #
id-inCs2ScfToSsfGeneric             OBJECT IDENTIFIER ::= { id-contract 6 } #
id-inCs2ScfToSsfDpSpecific          OBJECT IDENTIFIER ::= { id-contract 7 } #
id-inCs2ScfToSsfTrafficManagement   OBJECT IDENTIFIER ::= { id-contract 8 } #
id-inCs2ScfToSsfServiceManagement   OBJECT IDENTIFIER ::= { id-contract 9 } #
id-inCs2SsfToScfServiceManagement   OBJECT IDENTIFIER ::= { id-contract 10 } #
id-inCs2ScfToSsfStatusReporting     OBJECT IDENTIFIER ::= { id-contract 11 } #
id-inCs2ScfToSsfTriggerManagement   OBJECT IDENTIFIER ::= { id-contract 12 } #

-- srf/scf Contracts
id-contract-srf-scf                 OBJECT IDENTIFIER ::= { id-contract 13 } #

-- SCF-SDF contracts --
id-contract-dap                     OBJECT IDENTIFIER ::= { id-contract 1 }
id-contract-dapExecute               OBJECT IDENTIFIER ::= { id-contract 2 }

-- SDF - SDF Contracts.
id-contract-indsp                   OBJECT IDENTIFIER ::= { id-contract 14 }
id-contract-shadowConsumer          OBJECT IDENTIFIER ::= { id-contract 15 }
id-contract-shadowSupplier          OBJECT IDENTIFIER ::= { id-contract 17 }

-- scf/scf Contracts
id-contract-scf-scf                 OBJECT IDENTIFIER ::= { id-contract 18 } #
id-contract-dssp                     OBJECT IDENTIFIER ::= { id-contract 19 } #

-- cusf/scf Contracts
id-contract-scf-cusf                OBJECT IDENTIFIER ::= { id-contract 20 } #
id-contract-cusf-scf                OBJECT IDENTIFIER ::= { id-contract 21 } #

-- ssf/scf Operation Packages
id-package-scfActivation              OBJECT IDENTIFIER ::= { id-package 11 } #
id-package-basicBCPDP                OBJECT IDENTIFIER ::= { id-package 12 } #
id-package-advancedBCPDP             OBJECT IDENTIFIER ::= { id-package 14 } #
id-package-srf-scfActivationOfAssist  OBJECT IDENTIFIER ::= { id-package 15 } #
id-package-assistConnectionEstablishment OBJECT IDENTIFIER ::= { id-package 16 } #
id-package-genericDisconnectResource  OBJECT IDENTIFIER ::= { id-package 17 } #
id-package-nonAssistedConnectionEstablishment OBJECT IDENTIFIER ::= { id-package 18 } #
id-package-connect                   OBJECT IDENTIFIER ::= { id-package 19 } #
id-package-callHandling              OBJECT IDENTIFIER ::= { id-package 20 } #
id-package-bcsmEventHandling         OBJECT IDENTIFIER ::= { id-package 21 } #
id-package-dpSpecificEventHandling   OBJECT IDENTIFIER ::= { id-package 22 } #

```

id-package-chargingEventHandling	OBJECT IDENTIFIER ::= {id-package 23}	#
id-package-ssfCallProcessing	OBJECT IDENTIFIER ::= {id-package 24}	#
id-package-scfCallInitiation	OBJECT IDENTIFIER ::= {id-package 25}	#
id-package-timer	OBJECT IDENTIFIER ::= {id-package 26}	#
id-package-billing	OBJECT IDENTIFIER ::= {id-package 27}	#
id-package-charging	OBJECT IDENTIFIER ::= {id-package 28}	#
id-package-trafficManagement	OBJECT IDENTIFIER ::= {id-package 29}	#
id-package-serviceManagementActivate	OBJECT IDENTIFIER ::= {id-package 30}	#
id-package-serviceManagementResponse	OBJECT IDENTIFIER ::= {id-package 31}	#
id-package-callReport	OBJECT IDENTIFIER ::= {id-package 32}	#
id-package-signallingControl	OBJECT IDENTIFIER ::= {id-package 33}	#
id-package-activityTest	OBJECT IDENTIFIER ::= {id-package 34}	#
id-package-statusReporting	OBJECT IDENTIFIER ::= {id-package 35}	#
id-package-cancel	OBJECT IDENTIFIER ::= {id-package 36}	#
id-package-cphResponse	OBJECT IDENTIFIER ::= {id-package 37}	#
id-package-entityReleased	OBJECT IDENTIFIER ::= {id-package 38}	#
id-package-triggerManagement	OBJECT IDENTIFIER ::= {id-package 39}	#
id-package-uSIHandling	OBJECT IDENTIFIER ::= {id-package 40}	#
id-package-facilityIEHandling	OBJECT IDENTIFIER ::= {id-package 41}	#
-- srf/scf Operation Packages		#
id-package-specializedResourceControl	OBJECT IDENTIFIER ::= { id-package 42}	#
id-package-srf-scfCancel	OBJECT IDENTIFIER ::= { id-package 43}	#
id-package-messageControl	OBJECT IDENTIFIER ::= { id-package 44}	#
id-package-scriptControl	OBJECT IDENTIFIER ::= { id-package 45}	#
-- SCF-SDF packages --		
id-package-search	OBJECT IDENTIFIER ::= {id-package 2}	
id-package-modify	OBJECT IDENTIFIER ::= {id-package 3}	
id-package-dapConnection	OBJECT IDENTIFIER ::= {id-package 10}	
id-package-execute	OBJECT IDENTIFIER ::= {id-package 4}	
-- SDF - SDF Packages.		
id-package-dspConnection	OBJECT IDENTIFIER ::= { id-package 47 }	
id-package-inchainedModify	OBJECT IDENTIFIER ::= { id-package 48 }	
id-package-inchainedSearch	OBJECT IDENTIFIER ::= { id-package 49 }	
id-package-chainedExecute	OBJECT IDENTIFIER ::= { id-package 50 }	
id-package-dispConnection	OBJECT IDENTIFIER ::= { id-package 51 }	
id-package-shadowConsumer	OBJECT IDENTIFIER ::= { id-package 52 }	
id-package-shadowSupplier	OBJECT IDENTIFIER ::= { id-package 53 }	
-- scf/scf Operation Packages		#
id-package-handlingInformation	OBJECT IDENTIFIER ::= {id-package 54}	#
id-package-notification	OBJECT IDENTIFIER ::= {id-package 55}	#
id-package-chargingInformation	OBJECT IDENTIFIER ::= {id-package 56}	#
id-package-userInformation	OBJECT IDENTIFIER ::= {id-package 57}	#
id-package-networkCapability	OBJECT IDENTIFIER ::= {id-package 58}	#
id-package-chainedSCFOperations	OBJECT IDENTIFIER ::= {id-package 59}	#
-- cusf/scf Operation Packages		#
id-package-emptyConnection	OBJECT IDENTIFIER ::= { id-package 60}	#
id-package-basic-cusf-scf	OBJECT IDENTIFIER ::= { id-package 61}	#
id-package-basic-scf-cusf	OBJECT IDENTIFIER ::= { id-package 62}	#
-- ssf/scf Abstract Syntaxes		#
id-as-ssf-scfGenericAS	OBJECT IDENTIFIER ::= {id-as 4}	#
id-as-ssf-scfDpSpecificAS	OBJECT IDENTIFIER ::= {id-as 5}	#
id-as-assistHandoff-ssf-scfAS	OBJECT IDENTIFIER ::= {id-as 6}	#
id-as-scf-ssfGenericAS	OBJECT IDENTIFIER ::= {id-as 7}	#
id-as-scf-ssfDpSpecificAS	OBJECT IDENTIFIER ::= {id-as 8}	#
id-as-scf-ssfTrafficManagementAS	OBJECT IDENTIFIER ::= {id-as 9}	#
id-as-scf-ssfServiceManagementAS	OBJECT IDENTIFIER ::= {id-as 10}	#
id-as-ssf-scfServiceManagementAS	OBJECT IDENTIFIER ::= {id-as 11}	#


```

id-as-scf-ssfStatusReportingAS          OBJECT IDENTIFIER ::= {id-as 12}          #
id-as-scf-ssfTriggerManagementAS       OBJECT IDENTIFIER ::= {id-as 13}          #

-- srf/scf Abstract Syntaxes                                                    #
id-as-basic-srf-scf                     OBJECT IDENTIFIER ::= { id-as 14}        #
id-as-basic-scf-srf                     OBJECT IDENTIFIER ::= { id-as 15}        #

-- SCF-SDF - abstract syntaxes --
id-as-indirectoryOperationsAS           OBJECT IDENTIFIER ::= {id-as 1}
id-as-indirectoryBindingAS              OBJECT IDENTIFIER ::= {id-as 2}
id-as-inExtendedDirectoryOperationsAS   OBJECT IDENTIFIER ::= {id-as 3}
id-as-inSESEAS                           OBJECT IDENTIFIER ::= {id-as 25}        #

-- SDF-SDF - abstract syntaxes
id-as-indirectorySystemAS                OBJECT IDENTIFIER ::= { id-as 16 }
id-as-indirectoryDSABindingAS            OBJECT IDENTIFIER ::= { id-as 17 }
id-as-indirectoryShadowAS                OBJECT IDENTIFIER ::= { id-as 18 }
id-as-indsaShadowBindingAS                OBJECT IDENTIFIER ::= { id-as 19 }

-- scf/scf Abstract Syntaxes                                                    #
id-as-scf-scfOperationsAS                OBJECT IDENTIFIER ::= {id-as 20}        #
id-as-distributedSCFSystemAS             OBJECT IDENTIFIER ::= {id-as 21}        #
id-as-scf-scfBindingAS                   OBJECT IDENTIFIER ::= {id-as 22}        #

-- cusf/scf Abstract Syntaxes                                                  #
id-as-basic-cusf-scf                     OBJECT IDENTIFIER ::= { id-as 23}        #
id-as-basic-scf-cusf                     OBJECT IDENTIFIER ::= { id-as 24}        #

-- Object Identifiers for SDF-SDF interface.
-- useful definitions
in-ds          OBJECT IDENTIFIER ::= {ccitt recommendation q 1228 sdf-objects (10)}
id-avc         OBJECT IDENTIFIER ::= {in-ds 29}
id-aca         OBJECT IDENTIFIER ::= {in-ds 24}
id-soa         OBJECT IDENTIFIER ::= {in-ds 21}

-- Object Identifiers for SDF-SDF interface.

-- SDF Attributes
id-soa-methodRuleUse          OBJECT IDENTIFIER ::= {id-soa 1}
id-aca-prescriptiveACI        OBJECT IDENTIFIER ::= { id-aca 4 }
id-aca-entryACI               OBJECT IDENTIFIER ::= { id-aca 5 }
id-aca-subentryACI            OBJECT IDENTIFIER ::= { id-aca 6 }

-- SDF Attribute Value Contexts
id-avc-assignment             OBJECT IDENTIFIER ::= {id-avc 1}

END

```

5 . S S F - S C F インタフェース

#

6 . S S F - S R F インタフェース

#

7. S C F - S D F インタフェース

7.1 I N の S D F インタフェースにおける T T C 標準 J T - X 5 0 0 の再利用の導入

7.1.1 T T C 標準 J T - X 5 0 0 の概念と I N との整合

S C F - S D F 間インタフェースおよび S D F の内容を規定するために T T C 標準 J T - X 5 0 0 シリーズの標準を用いる。T T C 標準 J T - X 5 0 0 シリーズ標準の考え方の多くは直接 I N の環境でも使用できるが、ディレクトリで導入された考え方を正しく理解するために、用語レベルの整合が必要であり、本項はこれを目的としている。そこで、I N の環境であいまいな用語に絞って記述する。

S C F の構成において、S C F の一部であるサービスデータ管理部が S D F とのインタラクションを行なう。これがディレクトリユーザエージェント (D U A) の概念に対応する。ユーザ対応に S C F が S D F とのアソシエーションを設定する際、D U A のインスタンスが S L P I 内で生成される。またそのインスタンスは、アソシエーションが終了するときに消滅する。

S D F はデータベースへの要求に応えるエンティティである。この機能エンティティはディレクトリシステムエージェント (D S A) に対応する。S C F - S D F 間のアソシエーションが設定されると、D S A のインスタンスがアソシエーションの存続期間中生成される。

ディレクトリは複数 D S A / 複数 S D F の集合である。この集合は特定サービスあるいは各種のサービスで使用できる。ディレクトリの概念は I N におけるデータベースシステムのそれと同等のものである。

ディレクトリはまたデータの倉庫と見ることができる。I N のサービスでは様々なデータアクセスをユーザに提供する。情報はエントリという型で構成される。エントリは識別 (あるいは名前で特定) できる情報の集まりである。エントリがオブジェクトを表わす (すなわち、オブジェクトに関する主要な情報を含む) 場合は、オブジェクトエントリと呼ばれる。

オブジェクトとは、識別 (あるいは名前で特定) でき、かつデータベース内に情報を保持することが必要なものである。典型的な例はユーザである。オブジェクトは複数のエントリによって表現できる。オブジェクトを表現するために用いる個々の情報は属性である。これはエントリと関連する。

I N 環境では、D S A に含まれるデータの管理と運用はサービス提供者が行なう。したがって、サービス提供者が管理者の役割を担う。これは、T T C 標準 J T - X 5 0 0 で言う管理上の権威者に相当する。サービス提供者はセキュリティ手順 (認証およびアクセス制御) を実行する。

7.1.2 T T C 標準 J T - X 5 0 0 の限定されたサブセットの利用

T T C 標準 J T - X 5 0 0 標準の目的は、ディレクトリサービスを提供することであって、S C F - S D F 間インタフェースの記述ではない。T T C 標準 J T - X 5 0 0 の機能は I N C S - 2 で必要とする機能よりも広い。本項では、I N C S - 2 の範囲においてインプリメントする際に考慮しサポートすべきディレクトリ抽象サービスの側面を示す。また、サポートしていないパラメータを受信した場合の対応についても触れる。様々なパラメータの状態を表す手段としてプロファイリングを用いる。

一つのメッセージで送信するパラメータ数をできるだけ少なくすることは、信号トラフィック量および処理時間の点から重要である。したがって、信号送信時に必須でないパラメータは削除する。削除されたパラメータを受信した場合、受信側のエンティティによって認識されるべきであるが扱われるべきでない。これによって、3 版のディレクトリの記述に従って将来プロファイルの拡張が許容される。

便利かつ明確のため、このプロファイルは A S N . 1 サブタイピングファシリティーズを用いて定義されるが、プロトコルの仕様を記述しているのではない。これは単純にインプリメントにおいて送信すべきでないパラメータを示している。ただし、ディレクトリ抽象サービスの元の定義に適合している値をデコードできる受信側のエンティティのふるまいを変えるものではない。そうではあるが、サブタイピングによって除外された要素は無視されるべきである。

7.1.3 前提条件

INCS-2のためのディレクトリ抽象サービスのプロファイルを設計するために、いくつかの前提条件を用いる。これらの前提条件は7.3項から参照される。

- ・ 前提条件1：INCS-2で用いるディレクトリ抽象サービスは3版である。1988年版でのみ用いられているパラメータは無視される。将来の能力セットで必要になるかもしれない機能は、サポートされていなくても考慮だけはしておくべきである。
- ・ 前提条件2：INにおけるエリアスエントリは、オブジェクトに対して別の名称を提供する単なる手段であり、したがって必要な場合は参照されるべきである。
- ・ 前提条件3：SCF-SDF間のオペレーションは破棄できない。たとえオペレーションが非常に多くの時間を要したとしても、タイマが満了するため、破棄する必要がない。

7.2 SDF情報モデル

ITU-T勧告X.501は、ディレクトリによって提供されるサービスをサポートするために必要な一般的な情報モデルを提供する。INに適用する場合、一般的な情報モデルはITU-T勧告X.501の1章から7章に適合すべきである。ただし、その一部にはサポートされる必要がないものがある。たとえば、使用に関してローカルに決められるDIT内容規則である

本標準の対象外のものもある。INCS-2では考慮されていない事項である。従って、ITU-T勧告X.501における以下のものは適用されない。

- 16.2.3項の節f)、h)、i)
- 16.2.4項の節a)、Compare オペレーションは使用せず、代わりに search オペレーションを使用する。したがって、Compare 許可は FilterMatch 許可に置き換わる。

7.2.1 情報フレームワーク

INはINサービスの要求を満たすためにITU-T勧告X.501情報フレームワークに数多くの拡張を定義する。これらの拡張のみ以降の節で定義される。言及されない限り他要素の定義はITU-T勧告X.501第3版と同様である。

7.2.1.1 メソッド

各々のメソッドは、DSAの制御下で実行されるDAPオペレーションのシーケンスを示す。DUAは、DSAがメソッドを実行するために必要な全ての情報を提供する責任がある。DSAは、DUAに返される全ての情報を収集する責任がある。

文書化の目的のため、クラス定義に記述フィールドを追加することが推奨される。

&InputAttribute フィールドは、メソッドの実行の入力として提示される属性を示す。

&OutputAttribute フィールドは、メソッドの実行の出力として返される属性を示す。

&SpecificInput フィールドは、メソッドの実行の入力として使用される付加情報の構文を示す。

&SpecificOutput フィールドは、メソッドの実行の出力として使用される付加情報の構文を示す。

&id フィールドはメソッドを一意に示す。

```
METHOD ::= CLASS {
    &InputAttributes           ATTRIBUTE OPTIONAL,
    &SpecificInput             OPTIONAL,
    &OutputAttributes         ATTRIBUTE OPTIONAL,
    &SpecificOutput           OPTIONAL,
```

&description	PrintableString OPTIONAL,
&id	OBJECT IDENTIFIER UNIQUE}

WITH SYNTAX {

[INPUT ATTRIBUTES	&InputAttributes]
[SPECIFIC-INPUT	&SpecificInput]
[OUTPUT ATTRIBUTES	&OutputAttributes]
[SPECIFIC-OUTPUT	&SpecificOutput]
[BEHAVIOUR	&description]
ID	&id}

7.2.1.2 DIT METHOD Use

7.2.1.2.1 概要

DIT METHOD Use は、サブスキーマ管理主体 (administrative authority) によって提供される仕様であり、特定のオブジェクトクラスのエントリに使用されるメソッドタイプを規定する。

DIT METHOD Use の定義は、以下を含む：

- それが適用されるオブジェクトクラスのタイプの識別；
- オブジェクトクラスのエントリが格納される場合は常にオブジェクトクラスと関連づけられるべきメソッドタイプの識別；

特定のオブジェクトクラスの DIT METHOD Use の定義は、後続して定義されるいかなるサブクラスにも適用する。

7.2.1.2.2 DIT METHOD Use の仕様

DIT METHOD Use の抽象構文は、以下の ASN.1 タイプによって表現される：

```
DITMethodUse ::= SEQUENCE {
    objectClass          OBJECT-CLASS.&id,
    methods              [1] SET OF METHOD.&id }
```

7.2.1.2.1 節に記述された定義分と、上述の ASN.1 タイプ定義分の様々な要素の関係は以下の通りである：

- objectClass 要素は、DIT METHOD Use を適用するオブジェクトクラスを示す；
- methods 要素は、オブジェクトクラスのエントリが格納される場合は常にオブジェクトクラスと関連づけられるべきタイプを規定する；

特定のオブジェクトクラスの DIT METHOD Use の定義は、後続して定義されるいかなるサブクラスにも適用する。

METHOD-USE-RULE 情報オブジェクトクラスは、DIT METHOD Use ルールの文書化を容易にするために提供される：

```
METHOD-USE-RULE ::= CLASS {
    &objectClassType    OBJECT-CLASS.&id          UNIQUE,
    &Mandatory          METHOD }
WITH SYNTAX {
    OBJECT-CLASS TYPE    &objectClassType
    METHODS              &Mandatory }
```

特定のオブジェクトクラスの METHOD-USE-RULE の定義は、後続して定義されるいかなるサブクラスにも適用する。

7.2.2 基本アクセス制御

SCF / SDF インタフェースにおける INCS - 2 の要求条件をサポートするために、X.500 第 3 版のアクセス制御情報 (ACI) に対して、以下の拡張が必要とされる。ここでは、拡張部分のみが記述される。残りの部分については、X.500 シリーズ勧告第 3 版の記述を適用する。

7.2.2.1 ProtectedItems

ProtectedItems の定義は、以下のように拡張される：

```
ProtectedItems ::= SEQUENCE {
    entry [0] NULL OPTIONAL,
    allUserAttributeTypes [1] NULL OPTIONAL,
    attributeType [2] SET OF AttributeType OPTIONAL,
    allAttributeValues [3] SET OF AttributeType OPTIONAL,
    allUserAttributeTypesAndValues [4] NULL OPTIONAL,
    attributeValue [5] SET OF AttributeTypeAndValue OPTIONAL,
    selfValue [6] SET OF AttributeType OPTIONAL,
    rangeOfValues [7] Filter OPTIONAL,
    maxValueCount [8] SET OF MaxValueCount OPTIONAL,
    maxImmSub [9] INTEGER OPTIONAL,
    restrictedBy [10] SET OF RestrictedValue OPTIONAL,
    contexts [11] SET OF ContextAssertion OPTIONAL,
    entryMethods [30] SET OF MethodIDs OPTIONAL }
```

entryMethods は、プロテクションのレベルが適用されるべき特定のメソッドを示す。

```
MethodIDs ::= METHOD.&id
```

7.2.2.2 GrantsAndDenials

GrantsAndDenials の定義は、以下のように拡張される：

```
GrantsAndDenials ::= BIT STRING {
    --ProtectedItems の全ての要素と共に用いても良い。
    grantAdd (0),
    denyAdd (1),
    grantDiscloseOnError (2),
    denyDiscloseOnError (3),
    grantRead (4),
    denyRead (5),
    grantRemove (6),
    denyRemove (7),
```

-- entry 要素と共にのみ用いてもよい。

grantBrowse (8),
denyBrowse (9),
grantExport (10),
denyExport (11),
grantImport (12),
denyImport (13),
grantModify (14),
denyModify (15),
grantRename (16),
denyRename (17),
grantReturnDN (18),
denyReturnDN (19),

--ProtectedItems の entry を除いた全ての要素と共に用いても良い。

grantCompare (20),
denyCompare (21),
grantFilterMatch (22),
denyFilterMatch (23),

-- ProtectedItems の entryMethod 要素と共に用いても良い。

GrantExecuteMethod (30),
denyExecuteMethod (31) }

grantExecuteMethod は、ユーザがエントリに対して特定のメソッドを実行できることを意味する。

注 - grantExecuteMethod の許可が、エントリと属性に対する通常のアクセス制御機構をバイパスするか否かは、ネットワークオペレータに依存する。

denyExecuteMethod は、ユーザがエントリに対して特定のメソッドを実行できないことを意味する。

7.2.3 属性コンテキスト

7.2.3.1 基本サービスコンテキスト

基本サービスコンテキストは、その属性値が意味論的に有効である基本サービスを属性値と関連付ける。例えば、基本サービスコンテキストは、それが使用されうる基本サービスのタイプを示す I S D N アドレスと関連けられる。U P T の場合、このコンテキストは異なる基本サービスの登録アドレスの定義を可能とする。

```
basicServiceContext      CONTEXT ::= {  
    WITH SYNTAX          BasicService  
    ID                    id-avc-basicService }
```

```
BasicService ::= INTEGER {  
    telephony              (1),  
    faxGroup2-3           (2),  
    faxGroup4              (3),  
    teletexBasicAndMixed  (4),
```

teletexBasicAndProcessable	(5),
teletexBasic	(6),
syntaxBasedVideotex	(7),
internationalVideotex	(8),
telex	(9),
messageHandlingSystems	(10),
osiApplication	(11),
audioVisual	(12)}

もし提示された値の中のコンテキストの値（つまり、基本サービスの値）が蓄積された値の中のそれと同一であれば、提示された値が蓄積された値と一致したと見なされる。

7.2.3.2 回線識別コンテキスト

回線識別コンテキストは、その属性値が意味論的に有効である回線識別を属性値と関連付ける。例えば、回線識別コンテキストは、発信回線に依存したルーチングを提供するルーチング番号と関連付けられる。

```
lineIdentityContext      CONTEXT ::= {
    WITH SYNTAX          IsdnAddress
    ID                    id-avc-lineIdentity}
IsdnAddress ::= AddressString {ub-international-isdn-number}
```

7.2.3.3 割り当てコンテキスト

割り当てコンテキストは、その属性値が割り当てられた識別名（例えば、顧客番号あるいは顧客名）を属性値と関連付ける。例えば、使用可能な資源群が複数の値を有する属性でモデル化され、顧客が識別名で指名されると仮定すると、この割り当てコンテキストは（予約された）資源の状態とそれを使用している現在の顧客名を提供するために使用される資源と関連付けられる。

```
assignmentContext      CONTEXT ::= {
    WITH SYNTAX          DistinguishedName
    ID                    id-avc-assignment}
```

7.2.4 属性定義

7.2.4.1 DIT METHOD Use のオペレーショナル属性

methodUse オペレーショナル属性は、オブジェクトクラスとその全てのサブクラスとともに使用されるべきメソッドを示すために使用される：

```
methodUse ATTRIBUTE ::= {
    WITH SYNTAX          MethodUseDescription
    EQUALITY MATCHING RULE objectIdentifierFirstComponentMatch
    USAGE                directoryOperation
    ID                    id-soa-methodRuleUse }
```

```
MethodUseDescription ::= SEQUENCE {
```


identifier	OBJECT-CLASS.&id,
name	SET OF DirectoryString { ub-schema } OPTIONAL,
description	DirectoryString { ub-schema } OPTIONAL,
obsolete	BOOLEAN DEFAULT FALSE,
information	[0] SET OF METHOD.&id }

methodUse オペレーショナル属性の値の identifier 要素は、それが適用するオブジェクトクラスタイプのオブジェクト識別子である。id-oa-allObject-classTypes の値は、それが全てのオブジェクトクラスタイプに適用することを示す。

methodUse オペレーショナル属性の値の information 要素は、identifier によって識別されるオブジェクトクラスに関連付けられるメソッドタイプを示す。

D I T の各々のエントリは、多くても一つの methodUse オペレーショナル属性によって管理される。さらにエントリは、構造化されたオブジェクトクラスのスーパークラスに対して定義された、全ての methodUse オペレーショナル属性によってもまた管理される。

注 - これは、execute オペレーションを処理する前に、S D F は、エントリの構造化されたオブジェクトクラスのインヘリタンスチェーンに属する構造化されたオブジェクトクラスと関連する methodUse の属性をチェックすべきであることを示す。

methodRule 属性は構造化されたオブジェクトクラスと関連づけられるため、同一の構造化されたオブジェクトクラス上の全てのエントリは、D I T におけるそれらの位置を管理する D I T 構造規則に関わらず、またそれらの内容を管理する D I T 内容規則に関わらず、同一の Method Use Rule を持つ、ということになる。

7.3 S C F - S D F インタフェースプロトコル

7.3.1 情報タイプおよび共通手順

7.3.1.1 Common Arguments

IN-CommonArguments ::= CommonArguments (

```

WITH COMPONENTS {
    ...,
    serviceControls (IN-ServiceControls),
    aliasedRDNs      ABSENT})

```

serviceControls 要素は 7.3.1.2 節に記述される。

aliasedRDNs 要素はコンパチビリティのためだけに第 3 版で現れている。したがって、第 3 版のディレクトリのインプリメントでは常に省略すべきである (前提条件 1)。

7.3.1.2 ServiceControls

IN-ServiceControls ::= ServiceControls (

```

WITH COMPONENTS {
    ...,
    timeLimit      ABSENT,
    sizeLimit      ABSENT,
    scopeOfReferral ABSENT,
    attributeSizeLimit ABSENT})

```

timeLimit 要素は要求を実行するための最大時間を示す。これは T C のオペレーションタイムと冗長であるため、必要ではない。

sizeLimit および attributeSizeLimit はオブジェクトもしくは属性に関して、結果上のサイズ限界を設定する。一般的な要求（要求元がD S Aの構造を知らない）の場合には有用であるが、I Nの場合にはこのような制限は適用可能とは思えない。

7.3.1.3 Entry Information Selection

```
IN-EntryInformationSelection ::= EntryInformationSelection (
    WITH COMPONENTS {
        ...,
        infoTypes          (attributeTypesAndValues))
```

attributes 要素は参照系のサービスで返送される属性を表す。allUserAttributes オプションは必要以上のトラフィックを生むために、サービス規定者には使用を避けるように忠告はするが、許されている。なるべく要求する属性の名前を指定する select のオプションを使用すべきである。

infoTypes 要素は属性の種別と値を返送すべきか、種別のみで良いかを規定する。I Nでは、サービスは主にその処理に関連する属性値を意識するため、本要素はデフォルト値を与えることで存在すべきではない。

7.3.1.4 EntryInformation

```
IN-EntryInformation ::= EntryInformation (
    WITH COMPONENTS {
        ...,
        fromEntry      (TRUE),
        information     (WITH COMPONENTS {
            ...,
            attributeType      ABSENT})) OPTIONAL))
```

fromEntry 要素はコピーを返送するのかエントリ自身を返送するのかを表す。I N C S - 2ではコピーメカニズムを使用しないため（前提条件3）、この要素のデフォルト値のみが使用される。

information パラメータは返送される関連情報を含む。infoTypes 要素で規定した選択（7.3.1.3節参照）を考慮すると、attribute のオプションのみが使用されるべきである。

7.3.1.5 S P K Mトークンプロファイル

X . 5 1 1 第 3 版の Bind オペレーションは S P K Mセキュリティ手順の使用を規定することが可能である。この項では I N C S - 2 オペレーションで使用可能な S P K Mトークンを概説する。

```
IN-Context-Data ::= Context-Data #
    (WITH COMPONENTS { #
        ..., #
        channelId      ABSENT, #
        seq-number     ABSENT})) #
```

Context-Data はオプションとコンフィデンシャリティ、インテグリティ、そして1ウェイ認証機能アルゴリズム識別子を規定する。

Channel 識別子(channelId)は I N C S - 2 では唯一つのチャンネルを使用するため、要求されない。

seq-number パラメータはトークンの順番を示す。メッセージの順序付けは低位プロトコルレイヤで確保されることから I N C S - 2 では要求されない。 #

IN-Mic-Header ::= Mic-Header #
(WITH COMPONENTS { #
..., #
snd-seq ABSENT}) #

sed-seq パラメータはトークンの順番を示す。メッセージの順序付けは低位プロトコルレイヤで確保されることから I N C S - 2 では要求されない。 #

Mic-Header はトークン識別子、コンテキスト識別子そしてインテグリティアルゴリズム識別子を含む。 #

IN-Wrap-Header ::= Wrap-Header #
(WITH COMPONENTS { #
..., #
snd-seq ABSENT}) #

Wrap-Header はトークン識別子、コンテキスト識別子、インテグリティアルゴリズム識別子そしてコンフィデンシャルティアルゴリズム識別子を含むヘッダー情報を規定する。 #

IN-Del-Header ::= Del-Header #
(WITH COMPONENTS { #
..., #
snd-seq ABSENT}) #

Del-Header はトークン識別子、コンテキスト識別子そしてインテグリティアルゴリズム識別子を含む。 #

7.3.2 オペレーション

7.3.2.1 Bind オペレーション

in-DirectoryBind OPERATION ::= {
ARGUMENT DirectoryBindArgument
RESULT DirectoryBindResult
ERRORS {in-DirectoryBindError}
CODE in-opcode-in-bind}

7.3.2.2 Search オペレーション

in-Search OPERATION ::= {
ARGUMENT IN-SearchArgument
RESULT IN-SearchResult
ERRORS {nameError | in-ServiceError | securityError | attributeError |
referral}
CODE id-opcode-in-search}

search オペレーションは、所望のエントリを求めてD I Tを検索するために用いる。

IN-SearchArgument ::= SearchArgument

```
(WITH COMPONENTS {  
    ...、  
    searchAliases          (TRUE),  
    selection              (IN-EntryInformationSelection),  
    pagedResults          ABSENT,  
    extendedFilter        ABSENT,  
    COMPONENT OF IN-CommonArguments})
```

filter パラメータは検索空間からエントリを限定するために使用する。コンパチビリティの理由で extendedFilter パラメータがディレクトリの 1993 年版で追加されたが、送られるべきではない。filter パラメータのみを送るべきである。

searchAliases パラメータは(ベースオブジェクトを除く)検索空間で遭遇したエリアスを考慮すべきか否かを示す。I Nでは検索する際、エリアスは常に参照されるため、本パラメータはデフォルト値でのみ使用される。

selection パラメータはエントリ中のどの情報(例、種別と値)が要求されているかを表す。(7.3.1.3 節参照)

pagedResults パラメータはページ毎の結果を要求するために使用する。PagedResults パラメータはsearch オペレーションの結果をページフォーマットで表すために使用する。結果はS C Fが処理するため、本情報はI N C S - 2では必要ない。

エラーに関しては、abandoned はサポートされない。

IN-SearchResult ::= SearchResult

```
(WITH COMPONENTS {  
    ...、  
    searchInfo              (WITH COMPONENTS {  
        ...、  
        entries            (WITH COMPONENT (IN-EntryInformation)),  
        partialOutcomeQualifier (PartialOutcomeQualifier  
            (WITH COMPONENTS {  
                ...、  
                queryReference  
            ABSENT)))OPTIONAL}}))
```

entries パラメータはフィルタを満足するエントリを含む。

partialOutcomeQualifier パラメータはsearch オペレーションが完全に終了しなかったときに存在する。これには、検索オペレーションが終了しなかった理由、オペレーションが止まった箇所に関する情報が含まれる。

queryReference パラメータはページを付与した結果が要求される場合に使用するものであるため、必要ではない。

7.3.2.3 AddEntry オペレーション

```
in-AddEntry OPERATION ::= {  
    ARGUMENT      AddEntryArgument  
    RESULT        AddEntryResult  
    ERRORS        {nameError | in-ServiceError | securityError | attributeError  
                  | updateError | referral}  
    CODE          id-opcode-in-addEntry }
```

7.3.2.4 RemoveEntry オペレーション

```
in-RemoveEntry OPERATION ::= {  
    ARGUMENT      RemoveEntryArgument  
    RESULT        RemoveEntryResult  
    ERRORS        {nameError | in-ServiceError | securityError | updateError | referral}  
    CODE          id-opcode-in-removeEntry }
```

7.3.2.5 ModifyEntry オペレーション

```
in-ModifyEntry OPERATION ::= {  
    ARGUMENT      IN-ModifyEntryArgument  
    RESULT        IN-ModifyEntryResult  
    ERRORS        {nameError | in-ServiceError | securityError | attributeError  
                  | updateError | referral}  
    CODE          id-opcode-in-modifyEntry }
```

IN-ModifyEntryArgument ::= ModifyEntryArgument

```
(WITH COMPONENTS {  
    ...、  
    selection      (IN-EntryInformationSelection))}
```

selection パラメータは返送すべき属性と値を規定する。(7.3.1.3 節参照)

IN-ModifyEntryResult ::= ModifyEntryResult

```
(WITH COMPONENTS {  
    ...、  
    null           ,  
    information     Information  
                    (WITH COMPONENTS {  
                        ...、  
                        entry (IN-EntryInformation))})}
```

modifyEntry オペレーションによって参照すべき情報がない場合は、null 結果が返送される。それ以外の場合は、information 結果の entry 要素によって情報が返送される。IN CS - 2 では本要素は 7.3.1.4 節で規定される。

7.3.2.6 Execute オペレーション

Execute オペレーションは入力情報を使い事前に定義されたメソッドにより実行ステップを行う。そして

結果情報を返送する。それぞれのステップはアルゴリズム実行の DAP オペレーション (execute オペレーションであることが可能) もしくはデシジョンテストである。

個々の DAP オペレーションのパラメータは入力パラメータと直前のオペレーションの結果そして / または、メソッドに関連するアルゴリズムの出力から取得される。出力パラメータは個々のオペレーションの結果から取得される。execute オペレーションは atomic オペレーションであることが考慮される。

```
execute OPERATION ::= {
    ARGUMENT      ExecuteArgument
    RESULT        ExecuteResult
    ERRORS        {attributeError|nameError | serviceError|referral|
                  securityError | updateError | executionError}
    CODE          id-opcode-execute}
ExecuteArgument ::= OPTIONALLY-PROTECTED{
    SET(
        object          [0]Name,
        method-id       [1]METHOD.&id({SupportedMethods}),
        input-assertions [2]SEQUENCE OF SEQUENCE{
            type METHOD.&InputAttributes.&id({SupportedMethods}{@method-id}),
            values SET OF METHOD.&inputAttributes.&id({SupportedMethods}{@method-
id})OPTIONAL,
            valuesWithContext [0]SET OF SEQUENCE{
                value [0]METHOD.&InputAttributes.&id({SupportedMethods}{@method-
id})
            }
        }OPTIONAL,
        contextList     [1]1SET OF Context
    }OPTIONAL,
    }OPTIONAL,
    specific-input     [3]METHOD.&SpecificInput({SupportedMethods}{@method-id})OPTIONAL,
    COMPONENTS OF CommonArguments},
    DIRQOP.&dapModifyEntryArg-QOP{@qop}}
```

object フィールドはメソッドの実行される DIT のエントリを示す。

execute-id フィールドは SDF 内で実行されるメソッドを示す。

input-assertions フィールドはメソッド実行の入力として使用される属性値のセットを提供する。

specific-input フィールドはメソッドを形成するために SDF により要求される付加情報を示す。

```
ExecuteResult ::= OPTIONALLY-PROTECTED{
    SET(
        method-id       [1]METHOD.&id({SupportedMethods}),
        output-assertions [2]SEQUENCE OF SEQUENCE{
            type METHOD.&OutputAttributes.&id({SupportedMethods}{@method-id}),
            values SET OF METHOD.&OutputAttributes.&Type({SupportedMethods}{@method-id,@.type})OPTIONAL,
            }
        }OPTIONAL,
    }OPTIONAL,
```

```

valuesWithContext [0]SET OF SEQUENCE{
    value
[0]METHOD.&OutputAttributes.&Type({SupportedMethods}{@method-id,@.type})OPTIONAL,
    contextList [1]SET OF Context
}OPTIONAL,
}OPTIONAL,
specific-output [3]METHOD.&SpecificOutput({SupportedMethods}{@method-id})OPTIONAL,
COMPONENTS OF CommonResults},
DIRQOP.&dapModifyEntryRes-QOP{@qop}}
specific-output フィールドはメソッド実行の結果として返送された情報を含む。
output-assertions はメソッド実行の結果として返送された属性値を含む。

```

```
SupportedMethods METHOD ::= {...}
```

SupportedMethods のセットはインタフェースに対し定義されたメソッド全てを含む。その正確な内容はそれをサポートするサービスそしてネットワークプロバイダーの合意により、ローカルに決定される事項である。

7.3.2.7 in-directoryUnbind オペレーション

in-directoryUnbind オペレーションはUnbind手順に対し、クラス4オペレーション動作を提供することで、X.511 directoryUnbind オペレーションを置き換える。

```
in-directoryUnbind OPERATION ::= inEmptyUnbind
```

7.3.3 Errors

ディレクトリで定義されている優先順位ルールが適用される。abandoned, abandonFailed エラーは、INCS-2ではサポートされない(前提条件1および4)ため、考慮しない。

7.3.3.1 Bind エラー

```
in-DirectoryBindError ::= DirectoryBindError
```

```

(WITH COMPONENTS{
    ...,
    error(WITH COMPONENTS{
        securityError(SecurityProblem(1|2|7|10)),
        ServiceError(ServiceProblem(2))}))

```

SecurityProblem10 は供給されたSPKMトークンが不正であることを示す。

#

可能であれば、Bindエラーを理解するために有り得るエラー全てがサポートされるべきである。

7.3.3.2 Service エラー

```

in-ServiceError ERROR ::= {
    PARAMETER IN-ServiceErrorParameter
    CODE id-errcode-in-serviceError}
IN-ServiceErrorParameter ::= ServiceErrorParameter

```

```
(WITH COMPONENTS {
  problem(ServiceProblem (1 | 2 | 3 | 4 | 5 | 6 | 8 | 9 | 10 | 11 | 12)))
```

ページを付与した結果の使用に関する `invalidQueryReference` は送るべきではない。

7.3.3.4 execution エラー

`executionError` は `Execute` オペレーションが完了しない場合返送される。

```
executionError ERROR ::= {
  PARAMETER OPTIONALLY-PROTECTED{
    SET{
      problem [0]ExecutionProblem,
      COMPONENTS OF CommonResults},
    DIRQOP.&dirErrors-QOP{@dirqop}}
  CODE id-errcode-executionError}
```

```
ExecutionProblem ::= INTEGER{
  missingInputvalues(1),
  executionFailure(2)}
```

`executionProblem` は `execute` オペレーション失敗の理由を示す。

- ・ `missingInputValues` は実行されるメソッドに対し不正な入力情報を含む `input-values` フィールドで返送される。
- ・ `executionFailure` はメソッドが正しく完了されない場合返送される。これはメソッド内に含まれる DAP オペレーションの一つの失敗で引き起こされる。

7.4 プロトコルの概略

7.4.1 遠隔オペレーション

ITU-T 勧告 X.880 | ISO/IEC 9072-1 は、ディレクトリの仕様で定義されている様々なディレクトリプロトコルのような ROS ベースのアプリケーションプロトコルを規定する際に有用である情報オブジェクトクラスを定義している。これらのクラスは、後述の項で使用される。ITU-T 勧告 X.880 | ISO/IEC 9072-1 で提供される規定技術は、オブジェクト間の一般的なプロトコルを定義するために使用される。SS7 のアプリケーションレイヤプロトコルとして実現する場合、ITU-T 勧告 X.880 | ISO/IEC 9072-1 の概念は、ITU-T 勧告 Q.775 の SS7 の概念に対応する。

7.4.2 ディレクトリの ROS オブジェクトおよびコントラクト

ITU-T 勧告 X.519 | ISO/IEC 9594-5 は、ユーザがディレクトリサービスにアクセスするためのアクセスポイントであるディレクトリと DUA 間の抽象サービスを定義している。7.5 節はこの抽象サービスを IN で利用した場合のサブセットを定義している。

ROS オブジェクトの `dua` クラスは、このクラスのインスタンスでありコントラクト `dapContract` もしくは `dapExecuteContract` の起動側である DUA を規定している。このコントラクトは、ディレクトリの仕様ではディレクトリ抽象サービスと言われ、7.4.2 節で ROS ベースの情報オブジェクトとして規定される。

```
dua ROS-OBJECT-CLASS ::=
```



```

{
    INITIATES          {dapContract | dapExecuteContract}
    ID                  id-rosObject-dua
}

```

ROS オブジェクトの directory クラスは、ディレクトリ抽象サービスの提供側を記述する。この提供側は dapContract/dapExecuteContract の応答側である。

```

directory             ROS-OBJECT-CLASS ::=
{
    RESPONDS          {dapContract | dapExecuteContract}
    ID                  id-rosObject-directory
}

```

ディレクトリはさらに、DUA に対して特定のアクセスポイントをサポートする DSA によって表されるものとしてモデル化できる。インテリジェントネットワークにおいては、各 DSA はディレクトリに対するアクセスポイントとなる可能性がある。

directory オブジェクトは DSA の集まり（それぞれは SDF 内に存在）として表現される。directory を構成する各 DSA は dap-dsa クラスのインスタンスである。dap-dsa オブジェクトは dapContract/dapExecuteContract における応答側の役割を担う。

```

dap-dsa               ROS-OBJECT-CLASS ::=
{
    RESPONDS          {dapContract | dapExecuteContract}
    ID                  id-rosObject-dapDSA
}

```

本標準の将来の版では、様々な目的を達成するために、DSA 間でお互いに連携動作することも可能となるであろう。

7.4.3 DAP Contract およびパッケージ

dapContract はコントラクトというクラスの情報オブジェクトとして定義される。

```

dapContract          CONTRACT ::=
{
    CONNECTION        dapConnectionPackage
    INITIATOR CONSUMER OF {searchPackage | modifyPackage}
    ID                  id-contract-dap
}

```

DUA と DSA が異なる IN の物理エンティティにあるときは、このアソシエーションコントラクトは、IN ディレクトリアクセスプロトコル (DAP) と呼ばれる SS7 のアプリケーションレイヤプロトコルとして実現される。本プロトコルの SS7 のアプリケーションコンテキストという意味での定義は、本標準の 7.5.2.1 節に示す。dapContract は、dapConnectionPackage という connection package, および 2 つのオペレーションパッケージ、searchPackage, modifyPackage から成る。

dapExecuteContract は CONTRACT クラスの情報オブジェクトとして定義される。

```

dapExecuteContract   CONTRACT ::=
{
    CONNECTION        dapConnectionPackage

```

```

INITIATOR CONSUMER OF {searchPackage | modifyPackage|executePackage}
ID                               id-contract-dapExecute
}

```

dapExecuteContract は dapConnectionPackage というコネクションパッケージ、及び3つのオペレーションパッケージ serchPackage、 modifyPackage、 executePackage から成る。

connectionPackage、dapConnectionPackage という connection package は、コネクションパッケージクラスの情報オブジェクトとして定義される。このコネクションパッケージの bind オペレーション、すなわち directoryBind は、ITU - T 勧告 X . 5 1 1 | ISO / IEC 9 5 9 4 - 3 で定義されている。このコネクションパッケージの unbind オペレーション、すなわち in-directoryUnbind は 7.3.2.8 節に定義される。

```

dapConnectionPackage      CONNECTION-PACKAGE ::=
{
    BIND                    in-DirectoryBind
    UNBIND                  directoryUnbind
    ID                      id-package-dapConnection
}

```

オペレーションパッケージ searchPackage および modifyPackage は、OPERATION-PACKAGE クラスの情報オブジェクトとして定義される。これらのオペレーションパッケージのオペレーションは、ITU - T 勧告 X . 5 1 1 | ISO / IEC 9 5 9 4 - 3 で定義されている。ITU - T 勧告 X . 5 1 1 | ISO / IEC 9 5 9 4 - 3 では、さらにディレクトリへのアクセスをサポートする別のオペレーションを定義しているが、インテリジェントネットワークにおいては使用しない。

```

searchPackage              OPERATION-PACKAGE ::=
{
    CONSUMER INVOKES       {search}
    ID                     id-package-search}

```

```

modifyPackage              OPERATION-PACKAGE ::=
{
    CONSUMER INVOKES       {addEntry | removeEntry | modifyEntry}
    ID                     id-package-modify}

```

(注) これらのパッケージが A S E として実現される場合は、この規定で定義されるアプリケーションコンテキストの構築のために使用されるものであって、個々の A S E あるいは A S E の組み合わせに対するパフォーマンスの要求を許容する意図はない。

オペレーションパッケージ、executePackage は OPERATION-PACKAGE クラスの情報オブジェクトとして定義される。このオペレーションパッケージのオペレーションは 7.3.2.6 節に定義される。

```

executePackage              OPERATION-PACKAGE ::=
{
    CONSUMER INVOKES       {execute}
    ID                     id-package-execute}

```

D U A は dapContract/dapExecuteContract の起動側であるため、コントラクトのオペレーションパッケ

ージの消費側の役割を担う。これはこのコントラクトおよび S S 7 での実現において、D U A のみがオペレーションを発行できることを意味する。

7.5 ディレクトリプロトコル抽象構文

7.5.1 抽象構文

今回のディレクトリアクセスプロトコルの版は、3 つの抽象構文のサポートを必要とする。

- a) S C F および S D F 間の対話を確立するために必要であり、T T C 標準 J T - Q 7 7 3 で規定されている T C 対話制御プロトコルデータユニットの抽象構文である `dialogue-abstract-syntax`。
- b) `directoryBind` および `directoryUnbind` オペレーションの起動、およびその結果通知のためのプロトコルデータユニットを送信するための抽象構文。
- c) 7.3.3 で規定されているオペレーションパッケージに含まれるオペレーションの起動、およびその結果通知のためのプロトコルデータユニットを送信するための抽象構文。

2 番目の抽象構文の値が得られる A S N . 1 タイプは、I T U - T 勧告 X . 8 8 0 で定義されているパラメータ化されたタイプ、`Bind{}` および `Unbind{}` を用いて規定される。

最後の抽象構文の値が得られる A S N . 1 タイプは、T T C 標準 J T - Q 7 7 3 で定義されているパラメータ化されたタイプ、`TCMessage{}` を用いて規定される。

これらすべての抽象構文は、T T C 標準 J T - Q 7 7 3 にリストアップされている制約を付けた基本 A S N . 1 コード化規則に従って (最小に) コード化される。

7.5.1.1 D A P 抽象構文

7.4.3 節で規定したオペレーションパッケージを実現する複数のディレクトリ A S E は、`executePackage` を除き、一つの抽象構文 `directoryOperationsAbstractSyntax` を共有する。これは ABSTRACT-SYNTAX クラスの情報オブジェクトとして規定される。

```
inDirectoryOperationsAbstractSyntax      ABSTRACT-SYNTAX ::= {
    BasicDAP-PDUs
    IDENTIFIED BY          id-as-directoryOperationsAS}
BasicDAP-PDUs ::= TCAPMessage { {DAP-Invokable}, {DAP-Returnable} }
DAP-Invokable           OPERATION ::= {search | addEntry | removeEntry | modifyEntry}
DAP-Returnable          OPERATION ::= {search | addEntry | removeEntry | modifyEntry}
```

7.5.1.2 拡張 D A P 抽象構文

7.4.3 節で規定したオペレーションパッケージを実現する複数のディレクトリ A S E は、`executePackage` を含み、一つの抽象構文 `inExtendedDirectoryOperationsAbstractSyntax` を共有する。これは ABSTRACT-SYNTAX クラスの情報オブジェクトとして規定される。

```
inExtendedDirectoryOperationsAbstractSyntax  ABSTRACT-SYNTAX ::= {
    Extended-BasicDAP-PDUs
    IDENTIFIED BY          id-as-inExtendedDirectoryOperationsAS}
Extended-BasicDAP-PDUs ::= TCMessage { {Extended-DAP-Invokable}, {Extended-DAP-Returnable} }
Extended-DAP-Invokable OPERATION ::= {search|addEntry|removeEntry|modifyEntry|execute}
Extended-DAP-Returnable OPERATION ::= {search|addEntry|removeEntry|modifyEntry|execute}
```

7.5.1.3 D A P バインディング抽象構文

7.4.3 節で規定したコネクションパッケージは、別の抽象構文 `directoryBindingAbstractSyntax` を用いて実現される。これは ABSTRACT-SYNTAX クラスの情報オブジェクトとして規定される。

```
inDirectoryBindingAbstractSyntax  ABSTRACT-SYNTAX ::= {
    DAPBinding-PDUs
    IDENTIFIED BY                  id-as-directoryBindingAS}
DAPBinding-PDUs ::= CHOICE {
    bind                            Bind {directoryBind},
    unbind                          Unbind {directoryUnbind}}
```

7.5.1.4 S E S E 抽象構文

追加の抽象構文である `inSESEAbstractSyntax` は、7.5.2.1 節に定義されている `iNdirectoryAccessWith3seAC` において使用される。これは、ABSTRACT-SYNTAX クラス の情報オブジェクトとして定義される。

```
inSESEAbstractSyntax            ABSTRACT-SYNTAX ::= {
    SESEapdus{{spkmThreeWay},NoInvocationId}
    IDENTIFIED BY                {id-as-inSESEAS}}
SESEapdus は、 I T U - T 勧告 X . 8 3 2 からインポートされ、 spkmThreeWay は I T U - T 勧告 X . 5
1 9 からインポートされる。
```

7.5.2 ディレクトリアプリケーションコンテキスト

7.5.2.1 ディレクトリアクセスアプリケーションコンテキスト

`dapContract` は `iNdirectoryAccessAC` として実現される。このアプリケーションコンテキストは、APPLICATION-CONTEXT クラスの情報オブジェクトとして規定される。

```
iNdirectoryAccessAC  APPLICATION-CONTEXT ::= {
    CONTRACT                    dapContract
    DIALOGUE MODE               structured
    TERMINATION                 basic
    ABSTRACT SYNTAXES           {dialogue-abstract-syntax |
                                iNdirectoryOperationsAbstractSyntax |
                                iNdirectoryBindingAbstractSyntax}
    APPLICATION CONTEXT NAME    id-ac-directoryAccessAC}
```

3 ウェイの認証が要求された場合、`dapContract` は、`iNdirectoryAccessWith3seAC` により実現される。このアプリケーションコンテキストは、APPLICATION-CONTEXT クラスの情報オブジェクトとして規定される。

```
iNdirectoryAccessWith3seAC  APPLICATION-CONTEXT ::= {
    CONTRACT                    dapContract
    DIALOGUE MODE               structured
    TERMINATION                 basic
    ADDITIONAL ASE              {id-se-treewayse}}
```

```

ABSTRACT SYNTAXES          {dialogue-abstract-syntax |
                             inDirectoryOperationsAbstractSyntax |
                             inDirectoryBindingAbstractSyntax |
inSESEAbstractSyntax }
APPLICATION CONTEXT NAME    id-ac-indirectoryAccessWith3seAC }

```

7.5.2.2 拡張ディレクトリアクセスアプリケーションコンテキスト

dapExecuteContract は、inExtendedDirectoryAccessAC により実現される。このアプリケーションコンテキストは、APPLICATION-CONTEXT クラスの情報オブジェクトとして規定される。

```

inExtendedDirectoryAccessAC APPLICATION-CONTEXT ::= {
    CONTRACT                dapExecuteContract
    DIALOGUE MODE           1structured
    TERMINATION             basic
    ABSTRACT SYNTAXES      {dialogue-abstract-syntax |
                             inExtendedDirectoryOperationsAbstractSyntax |
                             inDirectoryBindingAbstractSyntax }
    APPLICATION CONTEXT NAME id-ac-inExtendedDirectoryAccessAC }

```

3 ウェイの認証が要求された場合、dapExecuteContract は、inExtendedDirectoryAccessAC により実現される。このアプリケーションコンテキストは、APPLICATION-CONTEXT クラスの情報オブジェクトとして規定される。

```

inExtendedDirectoryAccessWith3seAC APPLICATION-CONTEXT ::= {
    CONTRACT                dapExecuteContract
    DIALOGUE MODE           structured
    TERMINATION             basic
    ADDITIONAL ASE         {id-se-treewayse }
    ABSTRACT SYNTAXES      {dialogue-abstract-syntax |
                             inExtendedDirectoryOperationsAbstractSyntax |
                             inExtendedDirectoryBindingAbstractSyntax |
                             inSESEAbstractSyntax }
    APPLICATION CONTEXT NAME id-ac-inExtendedDirectoryAccessWith3seAC }

```

7.5.3 オペレーションコード

本標準で定義されているパッケージに含まれるオペレーションは、ITU - T 勧告 X . 5 1 9 で規定されており、オペレーションコードの割当も ITU - T 勧告 X . 5 1 9 からインポートされる。

7.5.4 エラーコード

本標準で定義されているパッケージに含まれるエラーは、ITU - T 勧告 X . 5 1 9 で規定されており、エラーコードの割当も ITU - T 勧告 X . 5 1 9 からインポートされる。

7.5.5 バージョンおよび拡張ルール

ディレクトリは分散する可能性があるため、2 つ以上のディレクトリアプリケーションエンティティが連携動作して要求に応えることもある。複数のディレクトリ A E は、異なるプロトコルバージョン数によって示されるかどうかは不明であるが、ディレクトリサービスのディレクトリ仕様の異なる版に適合してインプリメントされるかもしれない。そのバージョン数は、複数のディレクトリ A E を直接結合する 2 つの間で最上位の共通バージョン数に調整される。

7.5.5.1 バージョンの決定

アソシエーションを受諾した時、すなわち D A P を利用した結合時、調整されたバージョンは、D U A とそれに接続された D S A 間で交換されるプロトコルのポイント・ポイントの側面のみ影響を与えなければならない。当該対話上の後続の要求あるいは応答は、調整されたバージョンによって制約を受けない。

(注) D A P におけるポイント・ポイントの側面で、異なるプロトコルバージョンによって現在表されるものはない。

7.5.5.2 D U A 側

7.5.5.2.1 D U A 側での要求および応答処理

D U A が要求を行う際、サポートしている仕様の内最も高い版を用いることができる。要求の中の一つ以上の要素がクリティカルな場合、criticalExtensions パラメータ内に拡張の数を示さなければならない。

(注) CHOICE, ENUMERATED, 列挙の形態使用される INTEGER 型の中で置き換えられた拡張の情報が、以前の仕様の版に従ってインプリメントされている D S A で固有なオペレーションに対して本質的である場合は、拡張をクリティカルに指定することが推奨される。

応答を処理する際 D U A は、

- a) bit string 内でビット名割当はすべて無視する。
- b) ENUMERATED 型あるいは列挙の形態使用される INTEGER 型の中の未知の名前付きの数値は、それが SET あるいは SEQUENCE のオプション要素である場合、すべて無視する。
- c) SETs 中の未知のすべての要素, SEQUENCE の最後にある未知のすべての要素, SET あるいは SEQUENCE のオプション要素である CHOICES 中の未知のすべての要素, は無視する。

(注) インプリメントによってはディレクトリ PDU の追加要素をローカルオプションとして無視しても良い。特に、未知の名前付きの数値, SET あるいは SEQUENCE の必須要素内の未知の CHOICE は、オペレーションを無効にすることなく無視できる。そのような要素の具体化は今後の課題である。

- d) 未知の属性型および属性値を受信することを、プロトコル違反とは考えない。
- e) 未知の属性型および属性値をオプションでユーザに通知する。

7.5.5.2.2 D U A 側でのエラーハンドリングのための拡張ルール

未知の表示問題およびパラメータをもつ既知のエラータイプを処理する際、D U A は、

- a) 未知の問題およびパラメータを受信することを、プロトコル違反とは考えない。(すなわち、TC-U-REJECT の発行あるいは対話のアボートは行わない。)
- b) 追加エラー情報をオプションでユーザに通知する。

未知のエラータイプを処理する際、D U A は、

- a) 未知のエラータイプを受信することを、プロトコル違反とは考えない。(すなわち、TC-U-REJECT の発行あるいはアプリケーションアソシエーションのアボートは行わない。)
- b) エラーをオプションでユーザに通知する。

7.5.5.3 D S A側での要求処理

オペレーションを実行する D S A が、意味不明の criticalExtensions を検出した場合には、serviceError として unavailableCriticalExtension 表示を返送しなければならない。

(注) 一つ以上の 0 値を持つ criticalExtensions を受信した場合は、その値に対応する拡張が存在しないかあるいはクリティカルではないことを表す。criticalExtensions の中に 0 値が存在することは、APDU の対応する拡張が存在するかしないかとしては表せない。

一方、D U A からの要求を処理する際 D S A は、

- a) bit string 内で未知のビット名割当はすべて無視する。
- b) ENUMERATED あるいは列挙の形態使用される INTEGER 型の中の未知の名前付きの数値は、それが SET あるいは SEQUENCE のオプション要素である場合、すべて無視する。
- c) SETs の中の未知のすべての要素、SEQUENCE の最後にある未知のすべての要素、SET あるいは SEQUENCE のオプション要素である CHOICES の中の未知のすべての要素は無視する。

(注) インプリメントによってはディレクトリ PDU の追加要素をローカルオプションとして無視しても良い。特に、未知の名前付きの数値、SET あるいは SEQUENCE の必須要素内の未知の CHOICE は、オペレーションを無効にすることなく無視できる。そのような要素の具体化は、I N C S - 2 では規定されない。

7.6 コンフォーマンス

本節は、本規定に対するコンフォーマンスの要求条件を定義する。

7.6.1 S C Fによるコンフォーマンス

本規定に対するコンフォーマンスを要求する S C F のインプリメントは、7.6.1.1 節から 7.6.1.3 節で規定された要求条件を満足する必要がある。

7.6.1.1 ステートメントの要求条件

以下が記述されなければならない。

- a) コンフォーマンスが要求され、S C F が送信可能な iNdirectoryAccessAC アプリケーションコンテキストのオペレーション。
- b) コンフォーマンスが要求されるセキュリティレベル（無し、簡易、厳密）。
- c) コンフォーマンスが要求され、S C F が起動可能な I T U - T 勧告 X . 5 1 1 | I S O / I E C 9 5 9 4 - 3 の 7.3.1 節の表に列挙されている拡張。

7.6.1.2 静的な要求条件

S C F は、

- a) 7.5.2.1 節で抽象構文によって定義される iNdirectoryAccessAC アプリケーションコンテキストをサポートする能力をもつ。
- b) 7.6.1.1 節 c) にあるコンフォーマンスが要求された拡張に対して適合する。

7.6.1.3 動的な要求条件

S C F は、

- a) 18.1.6 節で定義されている「使用するサービスへのマッピング」に適合する。
- b) 7.5.5.2 節で定義されている拡張手順の規則に適合する。

7.6.2 SDFによるパフォーマンス

本規定に対するパフォーマンスを要求するSDFのインプリメントは、7.6.2.1節から7.6.2.3節で規定された要求条件を満足する必要がある。

7.6.2.1 ステートメントの要求条件

以下が記述されなければならない。

- a) パフォーマンスが要求されるアプリケーションコンテキスト。本勧告の現在の版では、iNdirectoryAccessACアプリケーションコンテキストに対するパフォーマンスのみが要求される。

(注) アプリケーションコンテキストはここに記述するもの以外は分割されない。特に特定のオペレーションに対するパフォーマンスは要求されない。

- b) パフォーマンスが要求されるセキュリティレベル(無し, 簡易, 厳密)。
- c) パフォーマンスが要求される属性型。また、構文DirectoryStringに基づく属性に対して、UNIVERSALSTRING選択のためのパフォーマンスが要求されるか否か。
- d) パフォーマンスが要求されるオブジェクトクラス。
- e) パフォーマンスが要求され、SDFが応答可能なITU-T勧告X.511 | ISO/IEC 9594-3の7.3.1節の表に列挙されている拡張。
- f) ITU-T勧告X.501 | ISO/IEC 9594-2の8.8節, およびITU-T勧告X.511 | ISO/IEC 9594-3の7.6, 7.8.2, 9.2.2節で定義されているコレクティブ属性に対して、パフォーマンスが要求されるか否か。
- g) ITU-T勧告X.511 | ISO/IEC 9594-3の7.6, 7.8.2, 9.2.2節で定義されているハイアラキカル属性に対して、パフォーマンスが要求されるか否か。
- h) パフォーマンスが要求される、ITU-T勧告X.501 | ISO/IEC 9594-2で定義されているオペレーショナル属性型および他のオペレーショナル属性型。
- i) ITU-T勧告X.511 | ISO/IEC 9594-3の7.7.1節で定義されている別名の返送に対して、パフォーマンスが要求されるか否か。
- j) ITU-T勧告X.511 | ISO/IEC 9594-3の7.7.6節で記述されているように、返送されたエントリ情報が完全であることを示すためのパフォーマンスが要求されるか否か。
- k) ITU-T勧告X.511 | ISO/IEC 9594-3の11.3.2節で記述されているように、補助オブジェクトクラスを特定して、値の追加および/あるいは削除を行うために、オブジェクトクラスの属性を変更するためのパフォーマンスが要求されるか否か。
- l) 基本アクセス制御に対してパフォーマンスが要求されるか否か。
- m) 簡易化アクセス制御に対してパフォーマンスが要求されるか否か。
- n) パフォーマンスが要求されるネームバインディング。
- o) ITU-T勧告X.501 | ISO/IEC 9594-2で定義されているように、SDFがコレクティブ属性の管理ができるか否か。
- p) コンテキストに対してパフォーマンスが要求されるか否か。

7.6.2.2 静的な要求条件

SDFは、

- a) 7.5.2.1節で抽象構文によって定義されている、パフォーマンスが要求されるアプリケーションコンテキストをサポートする能力をもつ。
- b) ITU-T勧告X.501 | ISO/IEC 9594-2で抽象構文によって定義されている情報の枠組みをサポートする能力をもつ。

- c) 抽象構文によって定義されている、パフォーマンスが要求される属性型をサポートする能力をもつ。
- d) 抽象構文によって定義されている、パフォーマンスが要求されるオブジェクトクラスをサポートする能力をもつ。
- e) 7.6.2.1 節でパフォーマンスが要求される拡張に適合する。
- f) コレクティブ属性に対してパフォーマンスが要求されるのであれば、ITU-T 勧告 X.511 | ISO/IEC 9594-3 の 7.6, 7.8.2, 9.2.2 節で定義されている関連の手順を実行する能力をもつ。
- g) ハイアラキカル属性に対してパフォーマンスが要求されるのであれば、ITU-T 勧告 X.511 | ISO/IEC 9594-3 の 7.6, 7.8.2, 9.2.2 節で定義されている関連の手順を実行する能力をもつ。
- h) パフォーマンスが要求されるオペレーショナル属性型をサポートする能力をもつ。
- i) 基本アクセス制御に対してパフォーマンスが要求されるのであれば、基本アクセス制御の定義に適合する ACI アイテムを保持する能力をもつ。
- j) 簡易化アクセス制御に対してパフォーマンスが要求されるのであれば、簡易化アクセス制御の定義に適合する ACI アイテムを保持する能力をもつ。

7.6.2.3 動的な要求条件

SDF は、

- a) 18.1.6 節で定義されている「使用するサービスへのマッピング」に適合する。
- b) 7.5.5.3 節で定義されている拡張手順の規則に適合する。
- c) 基本アクセス制御に対してパフォーマンスが要求されるのであれば、基本アクセス制御の手順に従って SDF 内の情報を保護する能力をもつ。
- d) 簡易化アクセス制御に対してパフォーマンスが要求されるのであれば、簡易化アクセス制御の手順に従って SDF 内の情報を保護する能力をもつ。

7.7 SCF - SDF インターフェースの ASN.1 モジュール

ASN.1 モジュールの以下のセットは、IN-CS-2 のための SCF - SDF インターフェースの定義である。これらは、インテリジェントネットワークのサポートのために要求されるディレクトリ規定への変更を全て含んでいる。

モジュールは、変更された型を使用するために、これらの変更により影響を受ける定義も含んでいる。

7.7.1 IN-CS2-SDF-InformationFramework モジュール

このモジュールは、IN-CS-2 の要求に合わせるため、ITU-T 勧告 X.501 (InformationFramework モジュール) への拡張を含んでいる。

IN-CS2-SDF-InformationFramework

```
{ ccitt recommendation q 1228 module(0) sdfInformationFramework(9) version1(0) }
```

DEFINITIONS::=

BEGIN

-- EXPORTS ALL --

-- 型と値は、ディレクトリ抽象サービスの IN プロファイル、ディレクトリアクセスプロトコル及びディレクトリ情報シャドウイングプロトコルを定義する ASN.1 モジュール内での使用のためエクスポートされる。

- このモジュール内で定義される型と値は、ディレクトリ規定の範囲を含む他のASN.1モジュール内
- での使用、及びディレクトリサービスのアクセスのためこれらを使用するであろう、他のアプリケーション
- ユンでの使用のためにエクスポートされる。他のアプリケーションは、その固有な目的でこれらを使用
- する可能性があるが、しかし、これはディレクトリサービスの維持又は改良のために必要な拡張と変更
- に制約を与えない。

IMPORTS

```

informationFramework, upperBounds, selectedAttributeTypes
    FROM UsefulDefinitions {joint-iso-ccitt ds(5) module(1) usefulDefinitions(0) 3}
ATTRIBUTE, OBJECT-CLASS, objectClass, aliasedEntryName
    FROM InformationFramework informationFramework
DirectoryString{ }, objectIdentifierFirstComponentMatch
    FROM SelectedAttributeTypes selectedAttributeTypes
ub-schema
    FROM UpperBounds upperBounds
id-soa-methodRuleUse
    FROM IN-CS2-object-identifiers
        { ccitt recommendation q 1228 module(0) in-cs2-object-identifiers(17) version1(0)}

```

-- 属性データ型 --

- 以下の情報オブジェクトセットの定義は、おそらく、標準化の側面に対して、又は、仕様に準拠したプ
- ロトコルのインプリメントに対して、異なっている。セットは、Attribute コンポーネントの値、Attribute Type
- Value コンポーネントの値、及び AttributeValueAssertion コンポーネントの宣言により、テーブル制約を
- 規定することが要求される。

SupportedAttributes ATTRIBUTE ::= {objectClass | aliasedEntryName , ...}

-- METHOD 情報オブジェクトクラス規定 --

```

METHOD ::= CLASS {
    &InputAttributes          ATTRIBUTE OPTIONAL,
    &SpecificInput            OPTIONAL,
    &OutputAttributes         ATTRIBUTE OPTIONAL,
    &SpecificOutput           OPTIONAL,
    &description              PrintableString OPTIONAL,
    &id                       OBJECT IDENTIFIER UNIQUE }

```

WITH SYNTAX {

```

[INPUT ATTRIBUTES          &InputAttributes]
[SPECIFIC-INPUT            &SpecificInput]
[OUTPUT ATTRIBUTES        &OutputAttributes]
[SPECIFIC-OUTPUT          &SpecificOutput]
[BEHAVIOUR                 &description]
ID                          &id }

```

DITMethodUse ::= SEQUENCE {

```

    objectClass              OBJECT-CLASS.&id,

```

```

        methods                [1]    SET OF METHOD.&id }
METHOD-USE-RULE ::= CLASS {
    &objectClassType          OBJECT-CLASS.&id    UNIQUE,
    &Mandatory                METHOD }
WITH SYNTAX{
    OBJECT-CLASS TYPE &objectClassType
    METHODS &Mandatory
-- attributes --
methodUse ATTRIBUTE ::= {
    WITH SYNTAX                MethodUseDescription
    EQUALITY MATCHING RULE    objectIdentifierFirstComponentMatch
    USAGE                      directoryOperation
    ID                         id-soa-methodRuleUse }
MethodRuleUseDescription ::= SEQUENCE {
    identifier                  OBJECT-CLASS.&id,
    name                       SET OF DirectoryString { ub-schema } OPTIONAL,
    description                 DirectoryString { ub-schema } OPTIONAL,
    obsolete                   BOOLEAN DEFAULT FALSE,
    information                 [0]    SET OF METHOD.&id }
END

```

7.7.2 IN-CS2-SDF-BasicAccessControl モジュール

本モジュールは、I Nの要求を満たすためにI T U - T 勧告 X . 5 0 1 (InformationFramework モジュール) に対してなされた拡張を含む。

IN-CS2-SDF-BasicAccessControl

```
{ ccitt recommendation q 1228 module(0) sdfBasicAccessControl(10) version1(0) }
```

DEFINITIONS ::=

BEGIN

-- EXPORTS All --

-- 本モジュールで定義されている型と値は、ディレクトリ規定に含まれる他のASN.1モジュールにおける利用のため、およびこれらの型と値を使用してディレクトリサービスにアクセスする他のアプリケーションの利用のためにエクスポートされる。他のアプリケーションはこれらの型と値を固有の目的に応じて利用するかもしれないが、そのことによって、ディレクトリサービスの維持・向上のために必要となる拡張と修正が抑制されるべきではない。

IMPORTS

```

informationFramework, upperBounds, selectedAttributeTypes, basicAccessControl, directoryAbstractService
    FROM          UsefulDefinitions          {joint-iso-ccitt ds(5) module(1) usefulDefinitions(0) 3}
ATTRIBUTE, AttributeType, AttributeTypeAndValue, SubtreeSpecification, ContextAssertion
    FROM          InformationFramework      informationFramework
id-aca-prescriptiveACI, id-aca-entryACI, id-aca-subentryACI,sdf-InformationFramework
    FROM          IN-CS2-object-identifiers
                { ccitt recommendation q 1228 modules(0) in-cs2-object-identifiers(17) version1(0) }

```

```

ub-tag
    FROM          UpperBounds          upperBounds
METHOD
    FROM          IN-CS2-SDF-InformationFramework sdf-InformationFramework
Filter
    FROM          DirectoryAbstractService    directoryAbstractService
NameAndOptionalUID, directoryStringFirstComponentMatch, DirectoryString{ }
    FROM          SelectedAttributeTypes      selectedAttributeTypes
MaxValueCount, RestrictedValue, AuthenticationLevel, Precedence
    FROM          BasicAccessControl          basicAccessControl
;
-- 類型 --
ACIItem ::= SEQUENCE {
    identificationTag      DirectoryString { ub-tag },
    precedence             Precedence,
    authenticationLevel    AuthenticationLevel,
    itemOrUserFirst        CHOICE {
        itemFirst          [0] SEQUENCE {
            protectedItems ProtectedItems,
            itemPermissions SET OF ItemPermission },
        userFirst          [1] SEQUENCE {
            userClasses     UserClasses,
            userPermissions SET OF UserPermission }}}

ProtectedItems ::= SEQUENCE {
    entry                  [0] NULL OPTIONAL,
    allUserAttributeTypes [1] NULL OPTIONAL,
    attributeType          [2] SET OF AttributeType OPTIONAL,
    allAttributeValues     [3] SET OF AttributeType OPTIONAL,
    allUserAttributeTypesAndValues [4] NULL OPTIONAL,
    attributeValue         [5] SET OF AttributeTypeAndValue OPTIONAL,
    selfValue              [6] SET OF AttributeType OPTIONAL,
    rangeOfValues          [7] Filter OPTIONAL,
    maxValueCount          [8] SET OF MaxValueCount OPTIONAL,
    maxImmSub              [9] INTEGER OPTIONAL,
    restrictedBy           [10] SET OF RestrictedValue OPTIONAL,
    contexts               [11] SET OF ContextAssertion OPTIONAL,
    entryMethods           [30] SET OF MethodIDs
OPTIONAL}

MethodIDs ::= METHOD.&id
UserClasses ::= SEQUENCE {
    allUsers              [0] NULL OPTIONAL,
    thisEntry             [1] NULL OPTIONAL,

```

```

name                [2]    SET OF NameAndOptionalUID OPTIONAL,
userGroup           [3]    SET OF NameAndOptionalUID OPTIONAL,
                    -- dn 要素は GroupOfUniqueNames のエントリの名前でなけれ
                    -- ばならない
subtree             [4]    SET OF SubtreeSpecification OPTIONAL}
ItemPermission ::= SEQUENCE {
    precedence       Precedence OPTIONAL,
                    -- デフォルトは ACIItem での優先
    userClasses      UserClasses,
    grantsAndDenials GrantsAndDenials }
UserPermission ::= SEQUENCE {
    precedence       Precedence OPTIONAL,
                    -- デフォルトは ACIItem での優先
    protectedItems  ProtectedItems,
    grantsAndDenials GrantsAndDenials }
GrantsAndDenials ::= BIT STRING {
    -- ProtectedItems のどんな要素とともに使われるかもしれない許可
    grantAdd         (0),
    denyAdd          (1),
    grantDiscloseOnError (2),
    denyDiscloseOnError (3),
    grantRead        (4),
    denyRead         (5),
    grantRemove      (6),
    denyRemove       (7),
    -- エントリ要素とともに使われるかもしれない許可
    grantBrowse      (8),
    denyBrowse       (9),
    grantExport       (10),
    denyExport        (11),
    grantImport       (12),
    denyImport        (13),
    grantModify       (14),
    denyModify        (15),
    grantRename       (16),
    denyRename        (17),
    grantReturnDN     (18),
    denyReturnDN      (19),
    -- エントリを除く ProtectedItems のいかなる要素とともに使われるかもしれない許可
    grantCompare      (20),
    denyCompare       (21),
    grantFilterMatch  (22),
    denyFilterMatch   (23),

```

```

-- ProtectedItems の entryMethod 要素とともに使われるかもしれない許可
grantExecuteMethod          (30),
denyExecuteMethod          (31) }

-- 属性 --
prescriptiveACI             ATTRIBUTE ::= {
    WITH SYNTAX              ACIItem
    EQUALITY MATCHING RULE   directoryStringFirstComponentMatch
    USAGE                    directoryOperation
    ID                       id-aca-prescriptiveACI }
entryACI                    ATTRIBUTE ::= {
    WITH SYNTAX              ACIItem
    EQUALITY MATCHING RULE   directoryStringFirstComponentMatch
    USAGE                    directoryOperation
    ID                       id-aca-entryACI }
subentryACI                 ATTRIBUTE ::= {
    WITH SYNTAX              ACIItem
    EQUALITY MATCHING RULE   directoryStringFirstComponentMatch
    USAGE                    directoryOperation
    ID                       id-aca-subentryACI }

END

```

7.7.3 IN-CS2-SCF-SDF-Operations モジュール

IN-CS2-SCF-SDF-Operations

{ccitt recommendation q 1228 module(0) scf-sdf-operations(11) version1(0) }

DEFINITIONS ::=

BEGIN

-- EXPORTS All --

-- 本モジュールで定義されている型と値は、INディレクトリ規定に含まれる他のASN.1モジュール
-- における利用のため、およびこれらの型と値を使用してINディレクトリサービスにアクセスする他の
-- アプリケーションの利用のためにエクスポートされる。他のアプリケーションはこれらの型と値を固有
-- の目的に応じて利用するかもしれないが、そのことによって、ディレクトリサービスの維持・向上のた
-- めに必要な拡張と修正が抑制されるべきではない。

IMPORTS

```

informationFramework, distributedOperations, authenticationFramework, upperBounds,
directoryAbstractService, enhancedSecurity
    FROM UsefulDefinitions {joint-iso-ccitt ds(5) module(1) usefulDefinitions(0) 3}
CONTEXT, Context, DistinguishedName, Name
    FROM InformationFramework informationFramework
OperationProgress, ReferenceType, Exclusions, AccessPoint, ContinuationReference
    FROM DistributedOperations distributedOperations
CertificationPath, SIGNED {}, SIGNATURE {}, AlgorithmIdentifier
    FROM AuthenticationFramework authenticationFramework
id-avc-assignment, contexts, ros-InformationObjects, sdf-InformationFramework

```

```

FROM          IN-CS2-object-identifiers
              { ccitt recommendation q 1228 modules(0) in-cs2-object-identifiers(17) version1(0) }
basicServiceContext, lineIdentityContext
FROM          IN-Contexts          contexts
Code, OPERATION, ERROR
FROM          Remote-Operations-Information-Objects ros-InformationObjects
inEmptyUnbind
FROM          IN-CS2-classes {ccitt recommendation q 1228 modules(0) in-cs2-classes(4)
version1(0)}
METHOD
FROM          IN-CS2-SDF-InformationFramework          sdf-InformationFramework
OPTIONALLY-PROTECTED{ }, DIRQOP
FROM          EnhancedSecurity          enhancedSecurity
CommonArguments, CommonResults, attributeError, nameError, serviceError, securityError, referral,
updateError
FROM          DirectoryAbstractService          directoryAbstractService
;
execute      OPERATION ::= {
ARGUMENT    ExecuteArgument
RESULT      ExecuteResult
ERRORS      { attributeError | nameError |
              serviceError | referral |
              securityError |
              updateError | executionError }
CODE        id-opcode-execute }
ExecuteArgument ::=  OPTIONALLY-PROTECTED
SET {
  object      [0] Name,
  method-id   [1] METHOD.&id({SupportedMethods}),
  input-assertions [2] SEQUENCE OF SEQUENCE {
  type        METHOD.&InputAttributes.&id({SupportedMethods}{ @method-id}),
  values      SET    OF    METHOD.&InputAttributes.&Type({SupportedMethods}{ @method-
id,@.type})
OPTIONAL,
  valuesWithContext [0] SET OF SEQUENCE {
  value          [0]    METHOD.&InputAttributes.&Type({SupportedMethods}{ @method-
id,@.type})
OPTIONAL,
  contextList    [1]    SET OF Context
} OPTIONAL
} OPTIONAL,
  specific-input [3] METHOD.&SpecificInput({SupportedMethods}{ @method-id})
OPTIONAL,

```

```

        COMPONENTS OF CommonArguments },
        DIRQOP.&dapModifyEntryArg-QOP{ @qop } }
ExecuteResult ::= OPTIONALLY-PROTECTED
SET {
    method-id [1] METHOD.&id({SupportedMethods}),
    output-assertions [2] SEQUENCE OF SEQUENCE {
        type METHOD.&OutputAttributes.&id({SupportedMethods}{ @method-id}),
        values SET OF
            METHOD.&OutputAttributes.&Type({SupportedMethods}{ @method-id,@.type}) OPTIONAL,
            ValuesWithContext [0] SET OF SEQUENCE {
                value [0]
                    METHOD.&OutputAttributes.&Type({SupportedMethods}{ @method-
id,@.type})
                    OPTIONAL,
                contextList [1] SET OF Context
            } OPTIONAL
        } OPTIONAL,
        specific-output [3] METHOD.&SpecificOutput({SupportedMethods}{ @method-id}) OPTIONAL,
        COMPONENTS OF CommonResults },
        DIRQOP.&dapModifyEntryRes-QOP{ @qop } }
SupportedMethods METHOD ::= { ... }
in-directoryUnbind OPERATION ::= inEmptyUnbind
assignmentContext CONTEXT ::= {
    WITH SYNTAX DistinguishedName
    ID id-avc-assignment }

executionError ERROR ::= {
    PARAMETER OPTIONALLY-PROTECTED {
        SET
            problem [0] ExecutionProblem ,
            COMPONENTS OF CommonResults },
        DIRQOP.&dirErrors-QOP{ @dirqop } }
    CODE id-errcode-executionError }
ExecutionProblem ::= INTEGER
    missingInputValues (1),
    executionFailure(2) }
-- オブジェクト識別子割り当て
-- エラーコード
id-errcode-executionError Code ::= local:10
-- オペレーションコード
id-opcode-execute Code ::=local:10

END

```


7.7.4 IN-CS2-SCF-SDF-Protocol モジュール

本節では、「IN-CS2-SCF-SDF-Protocol」というASN.1モジュールの形で、このディレクトリの仕様に含まれるすべてのASN.1の型および値を記述している。

```

IN-CS2-SCF-SDF-Protocol {ccitt recommendation q 1228 modules(0) in-scf-sdf-protocol(12) version1(0)}
DEFINITIONS ::=
BEGIN
-- EXPORTS All --
-- 本モジュールで定義されている型と値は、ディレクトリ規定に含まれる他のASN.1モジュールにお
-- ける利用のため、およびこれらの型と値を使用してディレクトリサービスにアクセスする他のアプリケ
-- ーションの利用のためにエクスポートされる。他のアプリケーションはこれらの型と値を固有の目的に
-- 応じて利用するかもしれないが、そのことによって、ディレクトリサービスの維持・向上のために必要
-- となる拡張と修正が抑制されるべきではない。
IMPORTS
    directoryAbstractService , directorySecurityExchanges, protocolObjectIdentifiers
        FROM          UsefulDefinitions          ds-UsefulDefinitions
    ROS-OBJECT-CLASS, CONTRACT, OPERATION-PACKAGE, CONNECTION-PACKAGE, OPERATION
        FROM          Remote-Operations-Information-Objects          ros-InformationObjects
    Bind{ }, Unbind{ }
        FROM          Remote-Operations-Generic-ROS-PDUs          ros-genericPDUs
    TCMessage { }
        FROM          TCAPMessages          tc-Messages
    APPLICATION-CONTEXT, dialogue-abstract-syntax
        FROM          TC-Notation-Extensions          tc-NotationExtensions
    id-ac-indirectoryAccessAC, id-ac-inExtendedDirectoryAccessAC, id-rosObject-dua, id-rosObject-directory,
    id-rosObject-dapDSA,
    id-contract-dap, id-contract-dapExecute, id-package-dapConnection, id-package-search, id-package-modify,
    id-package-execute,
    id-as-indirectoryOperationsAS, id-as-inExtendedDirectoryOperationsAS, id-as-indirectoryBindingAS,
    id-as-inSESEAS, #
    id-ac-inExtendedDirectoryAccessWith3seAC, id-ac-indirectoryAccessWith3seAC, #
    ros-InformationObjects, ros-genericPDUs, tc-Messages, tc-NotationExtensions,
    sese-APDUs, #
    ds-UsefulDefinitions, scf-sdf-Operations
        FROM          IN-CS2-object-identifiers
                        {ccitt recommendation q 1228 modules(0) in-cs2-object-identifiers (17) version1 (0)}
    directoryBind, search, addEntry, removeEntry, modifyEntry
        FROM          DirectoryAbstractService          directoryAbstractService
    SESEapdus{ }, NoInvocationId #
        FROM          SeseAPDUs          sese-APDUs #
    spkmThreeWay #
        FROM          DirectorySecurityExchanges          directorySecurityExchanges #
    id-se-threewayse #

```

```

        FROM          ProtocolObjectIdentifiers          protocolObjectIdentifiers          #
execute, in-directoryUnbind
        FROM          IN-CS2-SCF-SDF-Operations          scf-sdf-Operations
;
-- アプリケーションコンテキスト --
inDirectoryAccessAC          APPLICATION-CONTEXT          ::=          {
        CONTRACT          dapContract
        DIALOGUE MODE          structured
        TERMINATION          basic
        ABSTRACT SYNTAXES          { dialogue-abstract-syntax |
                                inDirectoryOperationsAbstractSyntax |
                                inDirectoryBindingAbstractSyntax }
        APPLICATION CONTEXT NAME          id-ac-indirectoryAccessAC}
inDirectoryAccessWith3seAC          APPLICATION-CONTEXT          ::=          {          #
        CONTRACT          dapContract          #
        DIALOGUE MODE          structured          #
        TERMINATION          basic          #
        ADDITIONAL ASE          { id-se-threewayse }          #
        ABSTRACT SYNTAXES          { dialogue-abstract-syntax |          #
                                inDirectoryOperationsAbstractSyntax |          #
                                inDirectoryBindingAbstractSyntax |          #
                                inSESEAbstractSyntax }          #
        APPLICATION CONTEXT NAME          id-ac-indirectoryAccessWith3seAC}          #
inExtendedDirectoryAccessAC          APPLICATION-CONTEXT          ::=          {
        CONTRACT          dapExecuteContract
        DIALOGUE MODE          structured
        TERMINATION          basic
        ABSTRACT SYNTAXES          { dialogue-abstract-syntax |
                                inExtendedDirectoryOperationsAbstractSyntax |
                                inDirectoryBindingAbstractSyntax }
        APPLICATION CONTEXT NAME          id-ac-inExtendedDirectoryAccessAC}
inExtendedDirectoryAccessWith3seAC          APPLICATION-CONTEXT          ::=          {          #
        CONTRACT          dapExecuteContract          #
        DIALOGUE MODE          structured          #
        TERMINATION          basic          #
        ADDITIONAL ASE          { id-se-threewayse }          #
        ABSTRACT SYNTAXES          { dialogue-abstract-syntax |          #
                                inExtendedDirectoryOperationsAbstractSyntax |          #
                                inDirectoryBindingAbstractSyntax |          #
                                inSESEAbstractSyntax }          #
        APPLICATION CONTEXT NAME          id-ac-inExtendedDirectoryAccessWith3seAC}          #
-- ROSオブジェクト --
dua          ROS-OBJECT-CLASS          ::=          {

```

```

INITIATES          {dapContract|dapExecuteContract}
ID                 id-rosObject-dua}
directory ROS-OBJECT-CLASS ::= {
  RESPONDS        {dapContract|dapExecuteContract}
  ID              id-rosObject-directory}
dap-dsa  ROS-OBJECT-CLASS ::= {
  RESPONDS        {dapContract|dapExecuteContract}
  ID              id-rosObject-dapDSA}
-- コントラクト --
dapContract          CONTRACT ::= {
  CONNECTION         dapConnectionPackage
  INITIATOR CONSUMER OF {searchPackage | modifyPackage}
  ID                 id-contract-dap}
dapExecuteContract  CONTRACT ::= {
  CONNECTION         dapConnectionPackage
  INITIATOR CONSUMER OF {searchPackage | modifyPackage | executePackage}
  ID                 id-contract-dapExecute}
-- コネクションパッケージ --
dapConnectionPackage CONNECTION-PACKAGE ::= {
  BIND               directoryBind
  UNBIND             in-directoryUnbind
  ID                 id-package-dapConnection}
-- search and modify パッケージ
searchPackage        OPERATION-PACKAGE ::= {
  CONSUMER INVOKES  {search}
  ID                 id-package-search}
modifyPackage        OPERATION-PACKAGE ::= {
  CONSUMER INVOKES  {addEntry | removeEntry | modifyEntry}
  ID                 id-package-modify}
executePackage       OPERATION-PACKAGE ::= {
  CONSUMER INVOKES  {execute}
  ID                 id-package-execute}
-- 抽象構文 --
inDirectoryOperationsAbstractSyntax ABSTRACT-SYNTAX ::= {
  BasicDAP-PDUs
  IDENTIFIED BY     id-as-indirectoryOperationsAS}
BasicDAP-PDUs        ::= TCMMessage {{DAP-Invokable},{DAP-Returnable}}
DAP-Invokable        OPERATION ::= {search | addEntry | removeEntry | modifyEntry}
DAP-Returnable       OPERATION ::= {search | addEntry | removeEntry | modifyEntry}
inExtendedDirectoryOperationsAbstractSyntax ABSTRACT-SYNTAX ::= {
  Extended-BasicDAP-PDUs
  IDENTIFIED BY     id-as-inExtendedDirectoryOperationsAS}
Extended-BasicDAP-PDUs ::= TCMMessage {{Extended-DAP-Invokable},{Extended-DAP-Returnable}}

```

```

Extended-DAP-Invokable    OPERATION ::= {search | addEntry | removeEntry | modifyEntry | execute}
Extended-DAP-Returnable  OPERATION ::= {search | addEntry | removeEntry | modifyEntry | execute}
inDirectoryBindingAbstractSyntax    ABSTRACT-SYNTAX ::= {
    DAPBinding-PDUs
    IDENTIFIED BY    id-as-indirectoryBindingAS}
DAPBinding-PDUs ::= CHOICE {
    bind            Bind {directoryBind},
    unbind          Unbind {in-directoryUnbind}}
inSESEAbstractSyntax    ABSTRACT-SYNTAX ::= {                                #
    SESEapdus {{spkmThreeWay},NoInvocationId}                                #
    IDENTIFIED BY    {id-as-inSESEAS}}                                        #
END

```

8 . S D F - S D F インタフェース

8.1 I N T T C 標準 J T - X 5 0 0、D S P、D I S P サブセットへの導入

S D F - S D F インタフェースの目的は、1 つの S D F から別の S D F にサービプロファイルのコピーの転送を可能とし、データベース間のコピーを管理することである。T T C 標準 J T - X 5 0 0 の機能は I N C S - 2 の要求条件を満たすために必要な機能よりも広い。本項では、D S P および D I S P において、何を考慮しサポートすべきであるか、何を無視すべきであるかということを示す。様々なパラメータの状態を表す手段としてプロファイリングを用いる。

一つのメッセージで送信するパラメータ数をできるだけ少なくすることは、信号トラフィック量および処理時間を削減するために重要である。したがって、信号送信時に必須でないパラメータは削除する。信号を受信した際、削除されたパラメータは受信側のエンティティによって扱われるべきではないが、解釈は行われるべきである。これによって、1994 年版 T T C 標準 J T - X 5 0 0 のディレクトリの記述に従って将来プロファイルの拡張が許容される。

便利かつ明確のため、このプロファイルは A S N . 1 サブタイピングファシリティーズを用いて定義するが、これらの定義はプロトコルの仕様を記述しているのではない。これは単純にインプリメントにおいて送信すべきでないパラメータを示している。ただし、D S P および D I S P の本来の定義に適合している値をデコードできる受信側のエンティティのふるまいを変えるものではない。それでもなお、サブタイピングによって除外された要素は解釈されるべきであるが、扱われるべきではない。

8.2 前提条件

I N C S - 2 のための D S P , D I S P を設計するために、いくつかの前提条件を用いる。これらは以下の通り。

- ・ 前提条件 1: データの転送に関係する網オペレータ間の合意は、オフライン(例、管理オペレーション)で定義される。establishOperationalBinding オペレーションは合意を活性化するためだけに用いる。
- ・ 前提条件 2: 合意はオンラインのオペレーションでは変更できない。
- ・ 前提条件 3: terminateOperationalBinding オペレーションは、2 つの網オペレータ間の合意を終了するために用いる。これは、シャドウ消費側にあるコピーをもはや維持管理しないことを意味する。それは使用すべきではなく、削除すべきである。しかしながら、2 つの網間で今後のアソシエーション確立のために合意が要求されることもあるため、情報は保持すべきである。
- ・ 前提条件 4: シャドウの更新は、マスタコピーを管理しているシャドウ供給側から起動される。したがって、コピーの更新はシャドウされたコピーではなく、常にマスタコピーに対してなされる。更新の要求は、連鎖オペレーションを用いてマスタコピーに送信される。コピーは変化が生じる度に更新される。
- ・ 前提条件 5: D S A 間では直接的な参照のみが用いられる。したがって、オペレーションの連鎖は一度に限られる。一度連鎖してもオペレーションが実行できない場合は、リフェラルが返送されるべきである。
- ・ 前提条件 6: コピーのコピーを作成することはできない。コピーを得るためにはマスタコピーを参照するべきである。
- ・ 前提条件 7: シャドウするメカニズムは、特定の D A P オペレーションあるいは管理オペレーションによって起動される。管理オペレーションについては今後の検討課題である。
- ・ 前提条件 8: シャドウの合意が終了する時間はサービスの種別に依存する。ただし、ほとんどの場合はコピーの数に基づく。D I T のある一部に対してコピーの数が最大値に達すると、最も古いコピーを削除し、その合意を非活性化する必要がある。コピーの最大値は 1 となることもある。

- ・ 前提条件 9: S D F - S D F 間のオペレーションを廃棄することはできない。オペレーションの実行に時間がかかった場合は、タイマが満了するため、廃棄する必要はない。

8.3 I N T T C 標準 J T - X 5 0 0、D I S P サブセット

8.3.1 シャドウイング合意の規定

シャドウイング合意は以下のように規定される。

IN-ShadowingAgreementInfo ::= ShadowingAgreementInfo

(WITH COMPONENTS {

....

master ABSENT,

secondaryShadows ABSENT))

shadowSubject は、シャドウすべきサブツリー、エントリおよび属性を特定する。UnitOfReplication の要素は I T U - T 勧告 X . 5 2 5 の 9.2 節で定義されている。

updateMode は、シャドウ領域の更新をいつ実施するようにスケジュールするかを特定する。UpdateMode の要素は I T U - T 勧告 X . 5 2 5 の 9.3 節で定義されている。

master は、マスタ領域を含む D S A のアクセスポイントを含む。D S A は既にこの情報を認識しているため、I N では不要である。

secondaryShadows は、二次的なシャドウ情報がシャドウ供給側に後に供給されることを許容する。二次的なシャドウは I N の観点では無視する（前提条件 5）ため、本要素は含まれるべきではない。

8.3.2 DSA Shadow Bind

dSAShadowBind オペレーションは、シャドウを供給する期間の最初に用いる。

in-dSAShadowBind OPERATION ::= in-DirectoryBind

I N C S - 2 では T T C 標準 J T - Q 1 2 2 8 の 7.3.2.1 節に規定されているように in-DirectoryBind オペレーションを用いる。

8.3.3 IN-DSA Shadow Unbind

in-DSAShadowUnbind は unbind 手順のためにクラス 4 オペレーションの振る舞いを供給するための I T U - T 勧告 X . 5 2 5 dSAShadowUnbind オペレーションを置き換える。

in-dSAShadowUnbind OPERATION ::= inEmptyUnbind

8.3.4 Coordinate Shadow Update

inCoordinateShadowUpdate オペレーションは、シャドウ供給側が意図している更新の送信に関連するシャドウイング合意を指定するために用いる。

InCoordinateShadowUpdate OPERATION ::= {

ARGUMENT IN-CoordinateShadowUpdateArgument

RESULT IN-CoordinateShadowUpdateResult

ERRORS {shadowError}
CODE id-opcode-coordinateShadowUpdate}

IN-CoordinateShadowUpdateArgument ::= CoordinateShadowUpdateArgument (
WITH COMPONENTS {
....
updateStrategy (standard:{total|incremental})),

IN-CoordinateShadowUpdateResult ::= CoordinateShadowUpdateResult(
WITH COMPONENTS{
....
nullPRESERNT))

各パラメータの意味を以下に規定する。

- a) agreementID は、シャドウイング合意を特定する。
- b) lastUpdate は、当該の合意に対して最後の更新が送信された時刻に対するシャドウ供給側の認識を示し、シャドウ供給側の D S A によって提供された時刻である。このパラメータは、特定のシャドウの合意に対する最初の inCoordinateShadowUpdate あるいは inRequestShadowUpadte オペレーションの場合にのみ省略してもよい。
- c) updateStrategy は、シャドウ供給側がこの更新の際に用いようとしている更新の方法を特定する。I N C S - 2 においては、total あるいは incremental を使用すべきである。NoChanges の選択肢を用いることはない。
- d) securityParameters は、I T U - T 勧告 X . 5 1 1 | I S O | I E C 9 5 9 4 - 3 の 7.10 節で規定されている。

8.3.5 Update Shadow

inUpdateShadow オペレーションは、シャドウ供給側がシャドウ消費側に対して複製領域の更新を送信するために起動される。このオペレーションが起動される前に、inCoordinateShadowUpdate あるいは inRequestShadowUpdate オペレーションが、指定されたシャドウイング合意に対して正常に完了していなければならない。

inUpdateShadow OPERATION ::= {
ARGUMENT IN-UpdateShadowArgument
RESULT IN-UpdateShadowResult
ERRORS {shadowError}
CODE id-opcode-updateShadow}

IN-UpdateShadowArgument ::= UpdateShadowArgument (
WITH COMPONENTS {
....
updatedInfo (IN-RefreshInformation)),

```
IN-UpdateShadowResult ::= UpdateShadowUpdateResult(
    WITH COMPONENTS{
        ...,
        null                PRESENT}}
```

各パラメータの意味を以下に規定する。

- a) agreementID は、確立されているシャドウの合意を特定する。
- b) updateTime は、シャドウ供給側が供給する。この時刻は、次の inCoordinateShadowUpdate あるいは inRequestShadowUpdate オペレーションの際に、シャドウ供給側とシャドウ消費側がシャドウされた情報に関する共通認識を保証するために用いる。
- c) updateWindow は、それが存在する場合には、シャドウ供給側が更新を送信することを予定している次のウィンドウを示す。
- d) updatedInfo は、シャドウ消費側がシャドウされた情報を更新するために必要な情報を提供する。このパラメータで送信される情報の意味は、シャドウ消費側が供給された変更を反映することに行き着く。
- e) securityParameters は、ITU - T 勧告 X . 5 1 1 | ISO | IEC 9 5 9 4 - 3 の 7.10 節で規定されている。

```
IN-RefreshInformation ::= RefreshInformation (
    WITH COMPONENTS {
        ...,
        otherStrategy        ABSENT}}
```

各パラメータの意味を以下に規定する。

- a) noRefresh は、以前の要求から現時点までの間にシャドウされた情報に対して変更がなかったことを示している。これは updateShadow オペレーションが、シャドウイング合意 (updateMode) に規定されている特定の時間毎に送信されなければならないという場合で、かつ変更が実際には起きていない場合に用いられる。
- b) total は、シャドウされた情報の新しいインスタンスを提供する。しかしながら、信号量を削減するためには、差分更新の方法を用いることが好ましい。
- c) incremental は、シャドウされた情報を完全に置き換えるのではなく、最も最近に要求した inRequestShadowUpdate (あるいは inUpdateShadow の要求) の中の lastUpdate と現在のシャドウ更新の要求 (あるいは inRequestShadowUpdate の応答) の中の updateTime の間にシャドウされた情報に生じた変更だけを提供する。
- d) otherStrategy は、ディレクトリの仕様の範囲外のメカニズムを用いて更新を送信する能力を提供する。ただし IN C S - 2 では、全体変更あるいは差分更新の方法を使用すべきである。

要求が成功した場合、結果は送り返されてくるが、その中に情報は入っていない。

要求が失敗した場合、shadowError が報告される。特定のシャドウ問題が送り返される状況が ITU - T 勧告 X . 5 2 5 11.3.3 節で定義される。

8.3.6 Request Shadow Update

inRequestShadowUpdate オペレーションは、シャドウ消費側がシャドウ供給側からの更新を要求するために用いる。

```
inRequestShadowUpdate OPERATION ::= {  
  ARGUMENT          IN-RequestShadowUpdateArgument  
  RESULT  IN-RequestShadowUpdateResult  
  ERRORS  {shadowError}  
  CODE id-opcode-RequestShadowUpdate}
```

```
IN-RequestShadowUpdateArgument ::= RequestShadowUpdateArgument (  
  WITH COMPONENTS {  
    ...,  
    requestedStrategy          (standard:{incremental|total})),
```

```
IN-RequestShadowUpdateResult ::= RequestShadowUpdateResult(  
  WITH COMPONENTS {  
    ...,  
    null          PRESENT))
```

各パラメータの意味を以下に規定する。

- a) agreementID は、シャドウイング合意を特定する。
- b) lastUpdate は、最も最近に成功した更新においてシャドウ供給側が提供する時刻である。このパラメータは、特定のシャドウイング合意に対する最初の inCoordinateShadowUpdate あるいは inRequestShadowUpdate オペレーションの場合にのみ省略してもよい。
- c) requestedStrategy は、シャドウ消費側が要求する更新の種別を特定する。シャドウ消費側は、シャドウ供給側からの incremental あるいは total のいずれかを要求してよい。
- d) securityParameters は、ITU - T 勧告 X . 5 1 1 | ISO | IEC 9 5 9 4 - 3 の 7.10 節で規定されている。

8.4 IN TTC 標準 JT - X . 5 0 0、DSP サブセット

8.4.1 情報タイプおよび共通手順

8.4.1.1 Chaining Arguments

ChainingArguments は、全体の処理の中の一部をきちんと実行するのに必要な情報を D S A に転送するために、各 Chained オペレーションの中に存在する。

```
IN-ChainingArguments ::= ChainingArguments (  
  WITH COMPONENTS {  
    ...,  
    aliasDereferenced          ABSENT,  
    aliasedRDNs                ABSENT,
```

returnCrossRefs ABSENT,
info ABSENT,
timeLimit ABSENT,
excludeShadows ABSENT,
nameResolveOnMaster ABSENT))

各要素の意味を以下に規定する。

- a) originator 要素は、セキュリティパラメータの中で規定されていない場合に、要求元の名前を伝える。CommonArguments の中に requester が存在する場合は、このアーギュメントは省略してもよい。
- b) targetObject 要素は、ルーティングされるディレクトリエントリを持つオブジェクトの名前を伝える。このオブジェクトの役割は関連する特定のオペレーションに依存し、このオブジェクトのエントリが操作される場合もあり、また要求あるいはサブ要求に対して、複数のオブジェクトを含むベースオブジェクトとなることもある（例、ChainedModify）。この要素は、chained オペレーション内のオブジェクトまたはベースオブジェクトパラメータと同一の値をもつときに限り省略することができる。
- c) operationProgress 要素は、オペレーションの実行状況を D S A に通知するために使用される。したがって、全体のパフォーマンスの中で果たすべき役割を持つ。I N C S - 2 では、直接的な知識参照が前提となっているが、このパラメータは適用可能と思われる。というのは、オペレーションが連鎖されている S D F では、chained オペレーションの dsaReferral エラーを継続参照を用いて応答することが可能であるからである。
- d) traceInformation 要素は、連鎖の際に D S A 間でループすることを避けるために使用される。D S A は他の D S A にオペレーションを連鎖する前に、新しい要素を追跡情報に追加する。オペレーションの実行を要求された D S A は、追跡情報を調べることによって、オペレーションがループになっていないことをチェックする。
- e) aliasDereferenced 要素は、分散された名前解決の際に一つ以上の別名エントリに遭遇および展開したかどうかを示すために用いる論理値である。I N の別名エントリは、オブジェクトに対して別の名称を提供するだけの手段であるため、必要な場合は展開されるべきであり、したがってこの指標は必要ない。
- f) aliasedRDNs 要素は、targetObject 名の中の何個の R D N が、一つ（あるいはそれ以上）の別名エントリの aliasedEntryName 属性から生成されるかを示す。別名エントリに遭遇し展開されると、整数値が設定される。I N の別名エントリは、オブジェクトに対して別の名称を提供するだけの手段であるため、必要な場合は展開されるべきであり、したがってこの指標は必要ない。
- g) returnCrossRefs 要素は、分散オペレーションの実行中に使用された知識参照が、結果あるいはリフェラルとともにクロス参照として起動元 D S A に返送される必要があるかどうかを示すために使用される論理値である。I N C S - 2 では、直接的な知識参照が前提であるため、このパラメータは非適用と考えられる。
- h) referenceType 要素は、オペレーションの実行を要求された D S A に対して、その要求をそこへ中継するために用いられた知識の種別を示す。これによって D S A は、起動側が保持する知識の中にあるエラーを検出することができる。そのようなエラーが検出された場合は、invalidReference 問題を付与した ServiceError によって示されねばならない。ReferenceType の詳細は 8.4.1.3 節に記述する。
- i) info 要素は、1 つの共通の要求の処理に関連する複数の D S A 間で D M D (Directory Management Domain) 特有の情報を伝えるために使用される。I N C S - 2 では管理プロトコルは対象外であるため、このパラメータは非適用と考えられる。

- j) timeLimit 要素は、存在する場合は、オペレーションが完了すべき期限を示す。これはTCのオペレーションタイムと冗長であるため、不要である。
- k) securityParameters 要素は、ITU-T勧告X.511 | ISO | IEC 9594-3で規定されている。
- l) entryOnly 要素は、元のオペレーションが Search であり、subset を oneLevel に設定し、別名エントリが baseObject の直接の従属として遭遇した場合に、TRUE に設定される。targetObject 名の名前解決を実行し成功したDSAは、その名前の付いたエントリに対してのみオブジェクトの評価を実行せねばならない。
- m) authenticationLevel は、SDF間で実行された認証の方法を示すことが要求される際にオプションで提供される。authenticationLevel はITU-T勧告X.501 | ISO | IEC 9594-2で規定されている。
- n) uniqueIdentifier は、起動元の名前（起動元とは要求を中継したSDF）を確認することが要求される際にオプションで提供される。uniqueIdentifier はITU-T勧告X.501 | ISO | IEC 9594-2で規定されている。
- o) exclusions 要素は、Search オペレーションに対してのみ意味があり、存在する場合には、targetObject に従属するエントリのどのサブツリーを Search の結果から除外するかということを示す。
- p) excludeShadows 要素は、Search および List オペレーションに対してのみ意味があり、エントリのコピーではなくエントリに対して探索を適用すべきことを示す。このオプションの要素は、重複した結果を受信することを避けるための一つの方法として、DSAが使用することができる。INCS2では、直接的な知識参照が前提であるため、このパラメータは非適用と考えられる。
- q) nameResolveOnMaster 要素は、名前解決においてのみ意味があり、NSSR S (non-specific knowledge references)に遭遇したときのみ設定される。TRUE に設定された場合は、後続の名前解決、すなわち nextRDNTToBeResolved からの残りのRDNのマッチングは、エントリのコピーの情報は用いず、そのRDNによって指定されるエントリに対するマスタDSAにおいてなされるべきであることを示す。INCS-2では、直接的な知識参照が前提であるため、このパラメータは非適用と考えられる。

8.4.1.2 Chaining Results

Chaining Result は、各オペレーションの結果の中に存在し、オペレーションを起動したDSAに対するフィードバックを提供する。

```
IN-ChainingResults ::= ChainingResults (
WITH COMPONENTS {
...,
info ABSENT,
crossReferences ABSENT})
```

各要素の意味を以下に規定する。

- a) info 要素は、1つの共通の要求の処理に関連する複数のDSA間でDMD特有の情報を伝えるために使用される。INCS-2では管理プロトコルは対象外であるため、このパラメータは非適用と考えられる。
- b) crossReferences 要素は、対応する要求の returnCrossRefs 要素がTRUE に設定されていなければ、

Chaining Results 内には存在しない。I N C S - 2 では、直接的な知識参照が前提であるため、このパラメータは非適用と考えられる。

- c) securityParameters 要素は、I T U - T 勧告 X . 5 1 1 | I S O | I E C 9 5 9 4 - 3 で規定されている。これが存在しないことは、セキュリティパラメータのセットが存在しないことと等価と考えられる。
- d) alreadySearched 要素は、存在する場合は、targetObject の直下に従属するどの R D N が、ChainedSearch オペレーションの一部として処理され、したがって後続のサブ要求では除外されるべきであるか、ということを示す。

8.4.1.3 Reference Type

Reference Type の値は、I T U - T 勧告 X . 5 0 1 | I S O | I E C 9 5 9 4 - 2 で規定されている種々の参照の内の一つを示す。

IN-ReferenceType ::= ReferenceType (1|2|4|5|6|7|8)

I N C S - 2 では直接参照が前提であるため、値 3 (クロス参照) は非適用である。

8.4.1.4 Access Point 情報

Access Point には 3 つの型がある。

- a) Access Point の値は、ディレクトリ、特に D S A へのアクセスが行われる特定の点を識別する。Access Point は、関係する D S A の Name およびその No.7 信号方式で使用される D S A への PresentationAddress からなる。

```
IN-AccessPoint ::= AccessPoint (  
  WITH COMPONENTS {  
    ...,  
    protocolInformation      ABSENT})
```

address には、No.7 信号方式における D S A のネットワークアドレスが含まれる。

- b) MasterOrShadowAccessPoint の値は、ディレクトリへの Access Point を識別する。Access Point の category、すなわち master あるいは shadow は、名称コンテキストを示すか共通的に利用可能な複製領域を示すかに依存する。

```
IN-MasterOrShadowAccessPoint ::= MasterOrShadowAccessPoint (  
  WITH COMPONENTS {  
    ...,  
  COMPONENTS OF IN-AccessPoint})
```

- c) MasterAndShadowAccessPoints の値は、ディレクトリへの Access Point の集合、すなわち関連する D S A の集合を識別する。これらの Access Point は、エントリ情報を持つ D S A に対して、それぞれが、共通の名称コンテキストあるいは値が nonSpecific Knowledge 属性の値であるときに一つの D S A を主

とする名称コンテキストの共通集合から、参照するという性質を共有する。MasterAndShadowAccessPoints の値は、含まれる各 Access Point の値の category を示す。名称コンテキストのマスタ D S A の Access Point は、その集合に含まれる必要はない。

IN-MasterAndShadowAccessPoints ::= MasterOrShadowAccessPoint

AccessPointInformation の値は、ディレクトリへの一つ以上のアクセス点を特定する。

```
IN-AccessPointInformation ::= AccessPointInformation (  
WITH COMPONENTS {  
....  
COMPONENTS OF IN-MasterOrShadowAccessPoint))
```

8.4.1.5 Continuation Reference

Continuation Reference は、オペレーションの全体あるいは一部の実行を、一つ以上の異なった DSA でどのように継続できるかを示す。一般的には、関係する DSA が要求自体を伝達できない、あるいは伝達したくないときに、リフェラルとして返送される。

```
IN-ContinuationReference ::= ContinuationReference (  
WITH COMPONENTS {  
....  
aliasedRDNs          ABSENT,  
rdnsResolved         ABSENT,  
referenceType        (IN-ReferenceType),  
accessPoints         SET OF (IN-AccessPoint))
```

各要素の意味を以下に規定する。

- a) targetObject 名は、オペレーションを継続するときを使用することが提案される名前を示す。例えば、別名が展開されるかあるいは Search において基底オブジェクトが特定された場合には、入力要求で受信した targetObject 名と異なる場合がある。
- b) aliasedRDNs 要素は、targetObject 名の中の（もしあれば）いくつの RDN が、別名を展開することによって生成されたかを示す。IN の別名エントリは、オブジェクトに対して別の名称を提供するだけの手段であるため、必要な場合は展開されるべきであり、したがってこの指標は必要ない。
- c) operationProgress は、継続参照を受信した DSA あるいは DUA が、それに従いたい場合に、完了した名称解決の内、指定された DSA においてオペレーションの継続実行を支配する名称解決の量を示す。
- d) rdnsResolved 要素の値（名前の中のいくつかの RDN が、完全な名称解決の対象ではないが、クロス参照からは正しいと仮定された場合にのみ、存在する必要がある）は、内部参照のみを利用して、いくつの RDN が実際に解決したかを示す。INCS-2 では、直接的な知識参照が前提であるため、このパラメータは非適用と考えられる。
- e) referenceType 要素は、この継続を生成する際に利用された知識の種別を示す。
- f) accessPoints 要素は、この継続を実現するために参照すべきアクセス点を示す。不特定の従属参照が関係している場合にのみ、二つ以上の AccessPointInformation 項目が存在する可能性がある。

- g) entryOnly 要素は、元のオペレーションが Search であり、subset アーギュメントを oneLevel に設定し、別名エントリが baseObject の直接の従属として遭遇した場合に、TRUE に設定される。targetObject 名の名前解決を実行し成功した DSA は、その名前の付いたエントリに対してのみオブジェクトの評価を実行する。IN の別名エントリは、オブジェクトに対して別の名称を提供するだけの手段であるため、必要な場合は展開されるべきであり、したがってこの指標は必要ない。
- h) exclusions 要素は、受信側 DSA では調査すべきではない従属の名称コンテキストの集合を示す。
- i) returnToDUA 要素は、継続参照を生成した DSA が、途中の DSA を介して情報を返送したくなく（例えば、セキュリティの理由）、むしろ起動元 DUA と当該 DSA 間で DAP オペレーションによって直接情報を利用可能としたいことを示す場合に、オプションで提供される。returnToDUA が TRUE に設定されている場合は、referenceType は self に設定され得る。この要素は、IN においても網オペータ間で確立されるシャドウの合意をサポートするために使用可能である（例、在圏 S D F からホーム S D F への Modify が、アクセス制御の制約から失敗することもあり得る）。
- j) nameResolveOnMaster 要素は、継続参照を生成する DSA が NSSRs に遭遇したときにオプションで提供される。IN CS - 2 では、直接的な知識参照が前提であるため、このパラメータは非適用と考えられる。

8.4.2 DSA Bind

dSABind オペレーションは、ディレクトリサービスを提供する二つの DSA 間で協調する期間を開始するために用いる。

dSABind OPERATION ::= in-DirectoryBind

IN CS - 2 では、7.3.2.1 項に規定されているように in-DirectoryBind オペレーションを用いる。

8.4.3 IN DSA Unbind

in-DSAUnbind オペレーションは、ITU - T 勧告 X . 5 1 8 の dSAUnbind オペレーションの代わりに unbind 手続きのためのクラス 4 のオペレーションの振る舞いを提供する。

in-DSAUnbind OPERATION ::= inEmptyUnbind

8.4.4 連鎖オペレーション

DUA からオペレーションを受信した DSA は、そのオペレーションの連鎖形式を構築し、別の DSA に伝達することが可能である。IN CS - 2 では、連鎖形式のオペレーションを受信した DSA は、オペレーションを処理するか、起動 DSA が他網に存在するならば、受信側 DSA と同一網内の他の DSA と連鎖する。

連鎖形式のオペレーションを起動する DSA は、そのオペレーションのアーギュメントに署名することがオプションで可能である。そのように要求された場合、オペレーションを実行する DSA はオペレーションの結果に署名してもよい。

連鎖形式のオペレーションは、パラメータ化された型の IN-chained{} を用いて規定される。

```
IN-chained {OPERATION:operation} OPERATION ::= {
ARGUMENT                OPTIONALLY-PROTECTED {SET {
chainedArgument          (IN-ChainingArguments),
```

```

argument [0] operation.&ArgumentType}},
DIRQOP.&dspChainedOp-QOP@dirqop}
RESULT OPTIONALLY-PROTECTED {SET {
IN-chainedResult ABSENT,
result [0] operation.&ResultType}},
DIRQOP.&dspChainedOp-QOP@dirqop }
ERRORS {operation.&Errors EXCEPT(referral|dsaReferral)}
CODE operation.&code}

```

- a) IN-chainedArgument は、ChainingArguments の値であり、元々DUA が提供したアーギュメントに加えて、オペレーションを実行する DSA が実行するために必要な情報を含む。
- b) argument は、operation.&Argument の値であり、元々DUA が提供したアーギュメントからなる。

要求が成功した場合、オペレーション結果には以下の要素を含む。

- a) IN-chainedResult は、IN-ChainingResults の値であり、起動元 DUA に提供されるべき情報に加えて、連鎖における前位 DSA に必要とされる情報を含む。I N C S - 2 では、2 回以上の連鎖はない前提であるため、このパラメータの必要性はない。
- b) result は、operation.&Result の値であり、オペレーションの実行側から返送され、起動元 DUA へ返送されるべき結果で構成される。この情報は、I T U - T 勧告 X . 5 1 1 | I S O / I E C 9 5 9 4 - 3 で規定されている。

要求が失敗した場合には、operation.&Errors 集合の中のエラーの一つが返送される。ただし、dsaReferral が referral の代わりに返送される。

8.4.5 連鎖エラー

dsaReferral エラーは、DSA が何らかの理由でオペレーションを他の DSA に連鎖して実行を継続することを好まない場合に生成される。I N C S - 2 では、DSA は他網の DSA から受信したオペレーションでなければ連鎖することはないだろう。

```

IN-dsaReferral ERROR ::= dsaReferral (
WITH COMPONENTS {
....
reference (IN-ContinuationReference),
contextPrefix ABSENT})

```

各パラメータの意味を以下に規定する。

- a) IN-ContinuationReference は、起動側が要求をさらに他の DSA に伝達するために必要な情報を含む。
- b) このオペレーションに対する ChainingArguments の returnCrossRefs 要素が、TRUE の値であり、リフェラルが従属またはクロス参照に基づいている場合は、contextPrefix パラメータがオプションに含まれることもある。もしあれば、DSA の管理権限者はどの知識参照をこの方法で返送するかを決定する（例えば、その DSA に秘密の情報がある場合）。I N C S - 2 では、直接的な知識参照が前提であるため、このパラメータは非適用である。

8.5 プロトコルの概略

8.5.1 ROS オブジェクトおよびコントラクト

分散した DIB が存在する際に、ディレトリ抽象サービスを提供するために一般的に必要となる DSA 間の相互動作は、indspContract として定義される。このコントラクトに加わっている DSA は、dsp-dsa クラスの ROS オブジェクトとして定義される。本標準においては、このコントラクトは DSA 抽象サービスと称する。

```
dsp-dsa ROS-OBJECT-CLASS ::= {  
  BOTH {indspContract}  
  ID    id-rosObject-dspDSA }
```

シャドウ抽象サービスは、シャドウ供給側 DSA とシャドウ消費側 DSA 間の情報のシャドウイングを規定する。このサービスは二つの形式で表され、したがって二つの異なるコントラクトとして規定される。それらは ROS ベースの情報オブジェクトとして 8.5.2 項で規定される。

shadowConsumerContract は、シャドウ消費側、すなわち initiating-consumer-dsa クラスの ROS オブジェクトがコントラクトを起動するサービスの形式を表す。responding-supplier-dsa クラスの ROS オブジェクトが、このコントラクトにおいて応答する。

```
initiating-consumer-dsa ROS-OBJECT-CLASS ::= {  
  INITIATES {shadowConsumerContract}  
  ID    id-rosObject-initiatingConsumerDSA }
```

```
responding-supplier-dsa ROS-OBJECT-CLASS ::= {  
  RESPONDS {shadowConsumerContract}  
  ID    id-rosObject-respondingSupplierDSA }
```

shadowSupplierContract は、シャドウ供給側、すなわち initiating-supplier-dsa クラスの ROS オブジェクトがコントラクトを起動するサービスの形式を表す。responding-consumer-dsa クラスの ROS オブジェクトが、このコントラクトにおいて応答する。

```
initiating-supplier-dsa ROS-OBJECT-CLASS ::= {  
  INITIATES {shadowSupplierContract}  
  ID    id-rosObject-initiatingSupplierDSA }
```

```
responding-consumer-dsa ROS-OBJECT-CLASS ::= {  
  RESPONDS {shadowSupplierContract}  
  ID    id-rosObject-respondingConsumerDSA }
```

8.5.2 DSP コントラクトおよびパッケージ

indspContract は、CONTRACT クラスの情報オブジェクトとして定義される。


```

indspContract  CONTRACT ::= {
CONNECTION          dspConnectionPackage
INITIATOR CONSUMER OF {inchaindModifyPackage|inchaindSearchPackage|
                        chainedExecutePackage }
ID      id-contract-indsp}

```

異なるオープンシステムに存在する一組の DSA が相互作用するとき、このアソシエーションコントラクトは、No.7 信号方式のアプリケーションレイヤプロトコルとして実現され、INDIRECTOR システムプロトコル (DSP) と称する。No.7 信号方式のアプリケーションコンテキストという意味でのこのプロトコルの定義は、本標準の 8.6 項に示す。

indspContract は、コネクションパッケージ indspConnectionPackage および 3 つのオペレーションパッケージ inchaindModifyPackage, inchaindSearchPackage, chainedExecutePackage からなる。

コネクションパッケージ indspConnectionPackage は、CONNECTION-PACKAGE クラスの情報オブジェクトとして定義される。これはコネクションパッケージ indapConnectionPackage と同一である。

```

dspConnectionPackage  CONNECTION-PACKAGE ::= {
BIND  dSABind
UNBIND  in-DSAUnbind
ID      id-package-dspConnection}

```

オペレーションパッケージ inchaindModifyPackage と inchaindSearchPackage は、OPERATION-PACKAGE クラスの情報オブジェクトとして定義される。このパッケージのオペレーションは、ITU-T 勧告 X.518 で規定されている。

```

inchaindModifyPackage  OPERATION-PACKAGE ::= {
CONSUMER INVOKES      {chainedAddEntry|chainedRemoveEntry|
chainedModifyEntry}
ID      id-package-inchaindModify}

```

```

inchaindSearchPackage  OPERATION-PACKAGE ::= {
CONSUMER INVOKES      {chainedSearch}
ID      id-package-inchaindSearch}

```

オペレーションパッケージ chainedExecutePackage は、OPERATION-PACKAGE クラスの情報オブジェクトとして定義される。

```

chainedExecutePackage  OPERATION-PACKAGE ::= {
CONSUMER INVOKES      {chainedExecute}
ID      id-package-inchaindExecute}

```

indspContract においては、いずれの DSA が起動側になり、コントラクトのオペレーションを発行してもよ

い。

8.5.3 DISP コントラクトおよびパッケージ

shadowConsumerContract および shadowSupplierContract は、CONTRACT クラスの情報オブジェクトとして定義される。

```
shadowConsumerContract CONTRACT ::= {  
CONNECTION          dispConnectionPackage  
INITIATOR CONSUMER OF {shadowConsumerPackage}  
ID      id-contract-shadowConsumer}
```

```
shadowSupplierContract CONTRACT ::= {  
CONNECTION          dispConnectionPackage  
RESPONDER CONSUMER OF {shadowSupplierPackage}  
ID      id-contract-shadowSupplier}
```

I Nディレクトリ情報シャドウイングプロトコル (DISP) と称するシャドウ抽象サービスの二つの形式に対する No.7 信号方式での実現方法は、いくつかの No.7 信号方式のアプリケーションコンテキストという意味で本標準の 8.6 項に示す。

shadowConsumerContract および shadowSupplierContract は、共通のコネクションパッケージ dispConnectionPackage およびそれぞれに対応するオペレーションパッケージ shadowConsumerPackage あるいは shadowSupplierPackage からなる。

コネクションパッケージ dispConnectionPackage は、CONNECTION-PACKAGE クラスの情報オブジェクトとして定義される。これはコネクションパッケージ dapConnection Package と同一である。

```
dispConnectionPackage CONNECTION-PACKAGE ::= {  
BIND      dSAShadowBind  
UNBIND   in-dSAShadowUnbind  
ID      id-package-dispConnection}
```

オペレーションパッケージ shadowConsumerPackage および shadowSupplierPackage は、OPERATION-PACKAGE クラスの情報オブジェクトとして定義される。これらのパッケージのオペレーションは、I T U - T 勧告 X . 5 2 5 で規定されている。

```
shadowConsumerPackage OPERATION-PACKAGE ::= {  
CONSUMER INVOKES      {requestShadowUpdate}  
SUPPLIER INVOKES      {updateShadow}  
ID      id-package-shadowConsumer}
```

```
shadowSupplierPackage OPERATION-PACKAGE ::= {  
SUPPLIER INVOKES      {coordinateShadowUpdate|updateShadow}
```

ID id-package-shadowSupplier}

シャドウ消費側は、shadowConsumerContract の起動側であるため、shadowConsumer Package の消費側の役割を前提とする。すなわち、シャドウ消費側が requestShadowUpdate オペレーションを発行し、シャドウ供給側が updateShadow オペレーションを発行することを意味する。

シャドウ供給側は、shadowSupplierContract の起動側であるため、shadowSupplier Package の供給側の役割を前提とする。すなわち、シャドウ供給側がそのコントラクトのオペレーションを発行することを意味する。

8.6 プロトコル抽象構文

8.6.1 DSP 抽象構文

8.5.2 項で規定されているオペレーションパッケージを実現するディレクトリ ASE は、一つの抽象構文 inDirectorySystemAbstractSyntax を共有する。これは、ABSTRACT-SYNTAX クラスの情報オブジェクトとして定義される。

```
inDirectorySystemAbstractSyntax ABSTRACT-SYNTAX ::= {  
  BasicDSP-PDUs  
  IDENTIFIED BY id-as-indirectorySystemAS}  
  BasicDSP-PDUs ::= TCMMessage {{DSP-Invokable}, {DSP-Returnable}}  
  DSP-Invokable OPERATION ::= {chainedAddEntry | chainedRemoveEntry | chainedModifyEntry |  
    chainedSearch | chainedExecute}  
  DSP-Returnable OPERATION ::= {chainedAddEntry | chainedRemoveEntry | chainedModifyEntry |  
    chainedSearch | chainedExecute }
```

8.5.2 項で規定されたコネクションパッケージの実現については、別の抽象構文である inDirectoryDSABinding AbstractSyntax を使用する。これは、ABSTRACT-SYNTAX クラスの情報オブジェクトとして定義される。

```
inDirectoryDSABindingAbstractSyntax ABSTRACT-SYNTAX ::= {  
  DSABinding-PDUs  
  IDENTIFIED BY id-as-indirectoryDSABindingAS}  
  DSABinding-PDUs ::= CHOICE {  
    bind Bind {dSABind},  
    unbind Unbind {in-DSAUnbind}}
```

8.6.2 DISP 抽象構文

8.5.3 項で規定されているオペレーションパッケージを実現するディレクトリ ASE は、抽象構文 inDirectoryShadowAbstractSyntax を共有する。この抽象構文は、ABSTRACT-SYNTAX クラスの情報オブジェクトとして定義される。

```
inDirectoryShadowAbstractSyntax ABSTRACT-SYNTAX ::= {  
  BasicDISP-PDUs  
  IDENTIFIED BY id-as-indirectoryShadowAS}
```

```

BasicDISP-PDUs ::= TCMMessage {{DISP-Invokable}, {DISP-Returnable}}
DISP-Invokable OPERATION ::= {requestShadowUpdate|updateShadow|coordinateShadowUpdate}
DISP-Returnable OPERATION ::= {requestShadowUpdate|updateShadow|coordinateShadowUpdate}

```

上記に規定されたコネクションパッケージの実現については、別の抽象構文である inDirectoryDSAShadowBinding AbstractSyntax を使用する。これは、ABSTRACT-SYNTAX クラスの情報オブジェクトとして定義される。

```

inDirectoryDSAShadowBindingAbstractSyntax ABSTRACT-SYNTAX ::= {
DISPBinding-PDUs
IDENTIFIED BY id-as-indsaShadowBindingAS}

```

```

DISPBinding-PDUs ::= CHOICE {
bind Bind {dSAShadowBind},
unbind Unbind {in-DSAShadowUnbind}}

```

8.6.3 ディレクトリシステムアプリケーションコンテキスト

indspContract は、inDirectorySystemAC として実現される。このアプリケーションコンテキストは、APPLICATION-CONTEXT クラスの情報オブジェクトとして定義される。

```

InDirectorySystemAC APPLICATION-CONTEXT ::= {
CONTRACT dspContract
DIALOGUE MODE structured
TERMINATION basic
ABSTRACT SYNTAXES {dialogue-abstract-syntax|
inDirectorySystemAbstractSyntax|
inDirectoryDSABindingAbstractSyntax}
APPLICATION CONTEXT NAME id-ac-indirectorySystemAC}

```

3ウェイの認証が要求される場合は、indspContract は inDirectorySystemWith3seAC として実現される。このアプリケーションコンテキストは、APPLICATION-CONTEXT クラスの情報オブジェクトとして定義される。 #

```

inDirectorySystemWith3seAC APPLICATION-CONTEXT ::= { #
CONTRACT dspContract #
DIALOGUE MODE structured #
TERMINATION basic #
ADDITIONAL ASE {id-se-threewayse} #
ABSTRACT SYNTAXES {dialogue-abstract-syntax | #
inDirectorySystemAbstractSyntax| #
inDirectoryDSABindingAbstractSyntax | #
inSESEAbstractSyntax } #
APPLICATION CONTEXT NAME id-ac-indirectorySystemWith3seAC} #

```

8.6.4 ディレクトリシャドウアプリケーションコンテキスト

inshadowSupplierContract は、inshadowSupplierInitiatedAC として実現される。このアプリケーションコンテキストは、APPLICATION-CONTEXT クラスの情報オブジェクトとして定義される。

```
InshadowSupplierInitiatedAC APPLICATION-CONTEXT ::= {  
CONTRACT          shadowSupplierContract  
DIALOGUE MODE     structured  
TERMINATION       basic  
ABSTRACT SYNTAXES {dialogue-abstract-syntax|  
inDirectoryShadowAbstractSyntax|  
inDirectoryDSAShadowBindingAbstractSyntax}  
APPLICATION CONTEXT NAME id-ac-inShadowSupplierInitiatedAC}
```

3 ウェイの認証が要求される場合は、inshadowSupplierContract は inshadowSupplierInitiatedWith3seAC として #
実現される。このアプリケーションコンテキストは、APPLICATION-CONTEXT クラスの情報オブジェクト #
として定義される。 #

```
inshadowSupplierInitiatedWith3seAC APPLICATION-CONTEXT ::= { #  
CONTRACT          shadowSupplierContract #  
DIALOGUE MODE     structured #  
TERMINATION       basic #  
ADDITIONAL ASE    {id-se-threewayse} #  
ABSTRACT SYNTAXES {dialogue-abstract-syntax | #  
inDirectoryShadowAbstractSyntax | #  
inDirectoryDSAShadowBindingAbstractSyntax | #  
inSESEAbstractSyntax } #  
APPLICATION CONTEXT NAME id-ac-inShadowSupplierInitiatedWith3seAC} #
```

inshadowConsumerContract は、inshadowConsumerInitiatedAC として実現される。このアプリケーションコンテキストは、APPLICATION-CONTEXT クラスの情報オブジェクトとして定義される。

```
inshadowConsumerInitiatedAC APPLICATION-CONTEXT ::= {  
CONTRACT          shadowConsumerContract  
DIALOGUE MODE     structured  
TERMINATION       basic  
ABSTRACT SYNTAXES {dialogue-abstract-syntax|  
inDirectoryShadowAbstractSyntax|  
inDirectoryDSAShadowBindingAbstractSyntax}  
APPLICATION CONTEXT NAME id-ac-inShadowConsumerInitiatedAC}
```

3 ウェイの認証が要求される場合は、inshadowConsumerContract は inshadowConsumerInitiatedWith3seAC とし #
て実現される。このアプリケーションコンテキストは、APPLICATION-CONTEXT クラスの情報オブジェク #
トとして定義される。 #

```

inshadowConsumerInitiatedWith3seAC APPLICATION-CONTEXT ::= {                                     #
CONTRACT          shadowConsumerContract                                                         #
DIALOGUE MODE     structured                                                                     #
TERMINATION       basic                                                                           #
ADDITIONAL ASE    {id-se-threewayse}                                                            #
ABSTRACT SYNTAXES {dialogue-abstract-syntax|                                                    #
inDirectoryShadowAbstractSyntax|                                                                #
inDirectoryDSAShadowBindingAbstractSyntax|                                                    #
inSESEAbstractSyntax }                                                                           #
APPLICATION CONTEXT NAME id-ac-inShadowConsumerInitiatedWith3seAC}                             #

```

8.6.5 バージョンおよび拡張ルール

ディレクトリは分散する可能性があるため、2 つ以上のディレクトリアプリケーションエンティティが連携動作して要求に応えることもある。複数のディレクトリ A E は、プロトコルバージョン値が異なるかどうかは分からないが、異なるディレクトリサービス仕様の版に適合してインプリされているかもしれない。バージョン値は、直接結合関係にある 2 つのディレクトリ A E 間で共通の最も高い値に決められる。

8.6.5.1 バージョンの決定

D S P や D I S P を利用したアソシエーション、すなわち結合を受信した際、取り決めを行ったバージョンは、起動側 D S A とそれに接続された応答側 D S A 間で交換されるプロトコルのポイントトゥポイントの側面にのみ影響を与える。当該対話上の後続の要求あるいは応答は、取り決められたバージョンにより制約を受ける。

(注) D S P または D I S P におけるポイントトゥポイントの側面で、異なるプロトコルバージョンによって現在表されるものはない。

8.6.5.2 起動 D S A 側

8.6.5.2.1 起動 D S A 側での要求および応答処理

起動側 D S A が要求を行う際、サポートしている仕様の内最も高い版を用いることができる。要求の中の一つ以上の要素がクリティカルな場合、criticalExtensions パラメータ内に拡張の数を示さねばならない。

(注) CHOICE, ENUMERATED, (ENUMERATED として使用する) INTEGER 型の中で置き換えられた拡張の情報が、以前の仕様の版に従ってインプリメントされている応答側 D S A で固有なオペレーションに対して本質的である場合は、拡張をクリティカルに指定することが推奨される。

応答を処理する際、起動側 D S A は以下をせねばならない。

- a) bit string 内で未知のビット名割当はすべて無視する。
- b) ENUMERATED 型あるいは ENUMERATED として使用する INTEGER 型の中の未知の名前付きの数値は、それが SET あるいは SEQUENCE のオプション要素である場合、すべて無視する。
- c) SETs の中の未知のすべての要素, SEQUENCE の最後にある未知のすべての要素, SET あるいは SEQUENCE のオプション要素である CHOICES の中の未知のすべての要素, は無視する。

(注) インプリメントによってはディレクトリ PDU の追加要素をローカルオプションとして無視しても良い。特に、未知の名前付きの数値, SET あるいは SEQUENCE の必須要素内の未知の CHOICE は、オペレーションを無効にすることなく無視できる。そのような要素の具体化は今後の課題である。

- d) 未知の属性型および属性値を受信することを、プロトコル違反とは考えない。
- e) 未知の属性型および属性値をオプションでユーザに通知する。

8.6.5.2.2 起動D S A側でのエラー処理のための拡張ルール

未知の問題およびパラメータを持つ既知のエラータイプを処理する際、起動側D S Aは、以下をせねばならない。

- a) 未知の問題およびパラメータを受信することを、プロトコル違反とは考えない。(すなわち、TC-U-REJECTの発行あるいは対話のアポートは行わない。)
- b) 追加エラー情報をオプションでユーザに通知する。

未知のエラータイプを処理する際、起動側D S Aは、

- a) 未知のエラータイプを受信することを、プロトコル違反とは考えない。(すなわち、TC-U-REJECTの発行あるいはアプリケーションアソシエーションのアポートは行わない。)
- b) エラーをオプションでユーザに通知する。

8.6.5.3 応答D S A側での要求処理

オペレーションを実行する応答側D S Aが、意味不明のcriticalExtensions要素を検出した場合には、serviceErrorとしてunavailableCriticalExtension表示を返送しなければならない。

(注) 一つ以上の0値を持つcriticalExtensionsを受信した場合は、その値に対応する拡張が存在しないかあるいはクリティカルではないことを表す。criticalExtensionsの中に0値が存在することは、APDUの対応する拡張が存在するかしないかとしては表せない。

一方、起動側D S Aからの要求を処理する際、応答側D S Aは以下をせねばならない。

- a) bit string内で未知のビット名割当はすべて無視する。
- b) ENUMERATED型あるいはENUMERATEDとして使用するINTEGER型の中の未知の名前付きの数値は、それがSETあるいはSEQUENCEのオプション要素である場合、すべて無視する。
- c) SETsの中の未知のすべての要素、SEQUENCEの最後にある未知のすべての要素、SETあるいはSEQUENCEのオプション要素であるCHOICESの中の未知のすべての要素、は無視する。

(注) インプリメントによってはディレクトリPDUの追加要素をローカルオプションとして無視しても良い。特に、未知の名前付きの数値、SETあるいはSEQUENCEの必須要素内の未知のCHOICEは、オペレーションを無効にすることなく無視できる。そのような要素の具体化は今後の課題である。

8.7 コンフォーマンス

SDFのコンフォーマンスに関して、既に記述されているリストに加えて、以下の記述を追加すべきである。

8.7.1 SDFによるコンフォーマンス

8.7.1.1 ステートメントの要求条件

以下が記述されるべきである。

- a) コンフォーマンスが要求されるアプリケーションコンテキスト。本標準の現在の版では、inDirectorySystemAC アプリケーションコンテキストに対するコンフォーマンスが必要である。
注) アプリケーションコンテキストは、ここに記述されているものを除いて分割すべきではない。特に、コンフォーマンスは特定のオペレーションに対して要求されるべきではない。
- b) コンフォーマンスが inDirectorySystemAC アプリケーションコンテキストに対して要求される場合、ITU - T 勧告 X . 5 1 8 で規定されているオペレーションの連鎖モードがサポートされているか否か。
- c) コンフォーマンスが要求されるセキュリティレベル(無し, 簡易, 厳密)。
- d) コンフォーマンスが要求される属性型。さらに、構文 DirectoryString に基づく属性に対して、UNIVERSAL STRING 選択のためのコンフォーマンスが要求されるか否か。
- e) コンフォーマンスが要求されるオブジェクトクラス。
- f) ITU - T 勧告 X . 5 0 1 の 8.8 項および ITU - T 勧告 X . 5 1 1 の 7.6, 7.8.2, 9.2.2 項で規定されているコレクティブ属性に対して、コンフォーマンスが要求されるか否か。
- g) ITU - T 勧告 X . 5 1 1 の 7.6, 7.8.2, 9.2.2 項で規定されているハイアラキカル属性に対して、コンフォーマンスが要求されるか否か。
- h) コンフォーマンスが要求される、ITU - T 勧告 X . 5 0 1 で規定されているオペレーショナル属性型および他のオペレーショナル属性型。
- i) ITU - T 勧告 X . 5 1 1 の 7.7.1 項で規定されている別名の返送に対して、コンフォーマンスが要求されるか否か。
- j) ITU - T 勧告 X . 5 1 1 の 7.7.6 項で規定されている、返送されたエントリ情報が完全であることを示すためのコンフォーマンスが要求されるか否か。
- k) ITU - T 勧告 X . 5 1 1 の 11.3.2 項で規定されている、補助オブジェクトクラスを識別する値を追加および/あるいは削除するために、オブジェクトクラスの属性を変更するためのコンフォーマンスが要求されるか否か。
- l) 基本アクセス制御に対してコンフォーマンスが要求されるか否か。
- m) 簡易アクセス制御に対してコンフォーマンスが要求されるか否か。
- n) コンフォーマンスが要求される名前バインディング。
- o) ITU - T 勧告 X . 5 0 1 で規定されているコレクティブ属性を SDF が管理できるか否か。
- p) 属性コンテキストに対してコンフォーマンスが要求されるか否か。

8.7.1.2 静的な要求条件

SDFは、以下を満たさなければならない。

- a) 8.6 項で抽象構文によって定義されており、コンフォーマンスが要求されるアプリケーションコンテキストをサポートする能力をもつ。
- b) ITU - T 勧告 X . 5 0 1 で抽象構文によって定義されている情報の枠組みをサポートする能力をもつ。

- c) I T U - T 勧告 X . 5 1 8 で定義されている最小の知識の要求条件に適合する。
- d) 抽象構文によって定義されており、パフォーマンスが要求される属性型をサポートする能力をもつ。
- e) 抽象構文によって定義されており、パフォーマンスが要求されるオブジェクトクラスをサポートする能力をもつ。
- f) 前項においてパフォーマンスが要求されている拡張に適合する。
- g) コレクティブ属性に対してパフォーマンスが要求されるのであれば、I T U - T 勧告 X . 5 1 1 の 7.6, 7.8.2, 9.2.2 項で定義されている関連の手順を実行する能力をもつ。
- h) ハイアラキカル属性に対してパフォーマンスが要求されるのであれば、I T U - T 勧告 X . 5 1 1 の 7.6, 7.8.2, 9.2.2 項で定義されている関連の手順を実行する能力をもつ。
- i) パフォーマンスが要求されるオペレーショナル属性型をサポートする能力をもつ。
- j) 基本アクセス制御に対してパフォーマンスが要求されるのであれば、基本アクセス制御の定義に適合する A C I アイテムを保持する能力をもつ。
- k) 簡易アクセス制御に対してパフォーマンスが要求されるのであれば、簡易アクセス制御の定義に適合する A C I アイテムを保持する能力をもつ。

8.7.1.3 動的な要求条件

S D F は、以下を満たさなければならない。

- a) 18.1.7 項で定義されている「使用するサービスへのマッピング」に適合する。
 - b) I T U - T 勧告 X . 5 1 8 で定義されている、リフェラルに関するディレクトリの分散オペレーションに対する手順に適合する。
 - c) `directorySystemAC` アプリケーションコンテキストに対してパフォーマンスが要求されるのであれば、I T U - T 勧告 X . 5 1 8 で定義されている相互動作のリフェラルモードに適合する。
 - d) 相互動作の連鎖モードに対してパフォーマンスが要求されるのであれば、I T U - T 勧告 X . 5 1 8 で定義されている相互動作の連鎖モードに適合する。
- (注) この場合にのみ、D S A が `directorySystemAC` のオペレーションを発行可能であることが必要である。
- e) 7.5.5 項で定義されている拡張手順の規則に適合する。
 - f) 基本アクセス制御に対してパフォーマンスが要求されるのであれば、基本アクセス制御手順に従って S D F 内の情報を保護する能力をもつ。
 - g) 簡易アクセス制御に対してパフォーマンスが要求されるのであれば、簡易アクセス制御手順に従って S D F 内の情報を保護する能力をもつ。

8.7.2 シャドウ供給側によるパフォーマンス

シャドウ供給側の役割において、本ディレクトリ仕様に対するパフォーマンスを要求する S D F のインプリメントは、以下に規定する要求条件を満たす必要がある。

8.7.2.1 ステートメントの要求条件

以下が記述されるべきである。

- a) シャドウ供給側としてパフォーマンスが要求されるアプリケーションコンテキスト、すなわち `inShadowSupplierInitiatedAC` と `inShadowConsumerInitiatedAC`。
- b) パフォーマンスが要求されるセキュリティレベル（無し，簡易，厳密）。
- c) `UnitOfReplication` がどの程度までサポートされるか、特に、もしあれば以下のオプションのフィーチャの内いずれがサポートされるか、

- ObjectClass に関するエントリのフィルタリング
- AttributeSelection による属性の選択 / 除外
- 複製領域における従属知識の包含
- 従属知識に加えて拡張知識の包含

8.7.2.2 静的な要求条件

SDF は、以下を満たさなければならない。

- a) 前述の抽象構文によって定義されており、コンフォーマンスが要求されるアプリケーションコンテキストをサポートする能力をもつ。
- b) modifyTimestamp および createTimestamp オペレーショナル属性のサポートを提供する。

8.7.2.3 動的な要求条件

SDF は、以下を満たさなければならない。

- a) 18 章で定義されている使用するサービスへのマッピングに適合する。
- b) D I S P に関する I T U - T 勧告 X . 5 2 5 |ISO/IEC 9594-9 の手順に適合する。

8.7.3 シャドウ消費側によるコンフォーマンス

シャドウ消費側として、本ディレクトリ仕様に対するコンフォーマンスを要求する SDF のインプリメントは、以下に規定する要求条件を満たす必要がある。

8.7.3.1 ステートメントの要求条件

以下が記述されるべきである。

- a) シャドウ消費側としてコンフォーマンスが要求されるアプリケーションコンテキスト、すなわち inShadowSupplierInitiatedAC と inShadowConsumerInitiatedAC。
- b) コンフォーマンスが要求されるセキュリティレベル（無し，簡易，厳密）。
- c) SDF が重複する複製単位のシャドウイングをサポートするか否か。

8.7.3.2 静的な要求条件

SDF は、以下を満たさなければならない。

- a) 8.6 項で抽象構文によって定義されており、コンフォーマンスが要求されるアプリケーションコンテキストをサポートする能力をもつ。
- b) 重複する複製単位がサポートされる場合、modifyTimestamp および createTimestamp オペレーショナル属性のサポートを提供する。
- c) copyShallDo サービス制御のサポートを提供する。

8.7.3.3 動的な要求条件

SDF は、以下を満たさなければならない。

- a) 18.1.7 項で定義されている「使用するサービスへのマッピング」に適合する。
- b) D I S P に関する I T U - T 勧告 X . 5 2 5 |ISO/IEC 9594-9 の手順に適合する。

8.8 SDF - SDFインタフェースのASN.1モジュール

以下のASN.1モジュールのセットがIN CS-2のSDF - SDFインタフェースを定義する。本モジュールはインテリジェントネットワークをサポートするために要求されるディレクトリ仕様への全ての変更を含む。

本モジュールには変更された型が使用されているので、変更によって影響を受けた定義も含まれる。

8.8.1 IN-CS2-SDF-SDF-Protocol モジュール

本項は、IN-CS2-SDF-SDF-Protocol というASN.1モジュールの形で、このディレクトリ仕様に含まれる全てのASN.1型と値の定義を含む。

IN-CS2-SDF-SDF-Protocol

```
{ ccitt recommendation q 1228 module(0) in-cs2-sdf-sdf-Protocol(18) version1(0) }
```

DEFINITIONS ::=

BEGIN

-- EXPORTS All --

-- 本モジュールで定義される型と値は、ディレクトリ仕様内にある他のASN.1モジュールでこれらを使用できるようにするために、また、ディレクトリサービスにアクセスするために他のアプリケーションがこれらを使用できるようにするために、エクスポートされる。他のアプリケーションは自分自身の目的でこれらを使用してよいが、このことはディレクトリサービスを維持もしくは改良するために必要とされる拡張、変更を強要しない。

IMPORTS

```
distributedOperations, directoryShadowAbstractService, dsp , protocolObjectIdentifiers
```

```
FROM UsefulDefinitions ds-UsefulDefinitions
```

```
ROS-OBJECT-CLASS, CONTRACT, OPERATION-PACKAGE, CONNECTION-PACKAGE,  
Code,OPERATION
```

```
FROM Remote-Operations-Information-Objects ros-InformationObjects
```

```
Bind{ },Unbind{ }
```

```
FROM Remote-Operations-Generic-ROS-PDUs ros-genericPDUs
```

```
TCMessage { }
```

```
FROM TCAPMessages tc-Messages
```

```
APPLICATION-CONTEXT, dialogue-abstract-syntax
```

```
FROM TC-Notation-Extensions tc-NotationExtensions
```

```
dSABind,
```

```
chainedSearch, chainedAddEntry, chainedRemoveEntry, chainedModifyEntry, chained{ }
```

```
FROM DistributedOperations distributedOperations
```

```

dSAShadowBind,
coordinateShadowUpdate, updateShadow, requestShadowUpdate
    FROM DirectoryShadowAbstractService directoryShadowAbstractService

execute
    FROM IN-CS2-SCF-SDF-Operations scf-sdf-Operations

inEmptyUnbind
    FROM IN-CS2-classes { ccitt recommendation q 1228 modules(0) in-cs2-classes(4) version1(0) }

id-rosObject-dspDSA, id-rosObject-initiatingConsumerDSA, id-rosObject-respondingSupplierDSA,
id-rosObject-respondingConsumerDSA, id-rosObject-initiatingSupplierDSA,
id-contract-indsp, id-contract-shadowConsumer, id-contract-shadowSupplier,
id-package-dspConnection, id-package-inchainedModify, id-package-inchainedSearch,
id-package-chainedExecute,

id-package-dispConnection, id-package-shadowConsumer, id-package-shadowSupplier,
id-as-indirectorySystemAS, id-as-indirectoryDSABindingAS, id-as-indirectoryShadowAS,
id-as-indsaShadowBindingAS,
id-ac-indirectorySystemAC, id-ac-inShadowSupplierInitiatedAC, id-ac-inShadowConsumerInitiatedAC,
id-ac-inShadowSupplierInitiatedWith3seAC, id-ac-inShadowConsumerInitiatedWith3seAC,      #
id-ac-indirectorySystemWith3seAC,                                                    #
ds-UsefulDefinitions, ros-InformationObjects, ros-genericPDUs, tc-Messages,
tc-NotationExtensions, scf-sdf-Operations, scf-sdf-Protocol
    FROM IN-CS2-object-identifiers
        { ccitt recommendation q 1228 module(0) in-cs2-object-identifiers(17) version1(0) }

inSESEAbstractSyntax                                                                #
    FROM IN-CS2-SCF-SDF-Protocol scf-sdf-Protocol                                    #

id-se-threewayse                                                                    #
    FROM ProtocolObjectIdentifiers protocolObjectIdentifiers                       #

dspContract
    FROM DirectorySystemProtocol dsp

;

dsp-dsa ROS-OBJECT-CLASS ::= {
    BOTH          {indspContract}
    ID id-rosObject-dspDSA}

```

```

initiating-consumer-dsa ROS-OBJECT-CLASS ::= {
    INITIATES          {shadowConsumerContract}
    ID                  id-rosObject-initiatingConsumerDSA }

responding-supplier-dsa ROS-OBJECT-CLASS ::= {
    RESPONDS          {shadowConsumerContract}
    ID                  id-rosObject-respondingSupplierDSA }

initiating-supplier-dsa ROS-OBJECT-CLASS ::= {
    INITIATES          {shadowSupplierContract}
    ID                  id-rosObject-initiatingSupplierDSA }

responding-consumer-dsa ROS-OBJECT-CLASS ::= {
    RESPONDS          {shadowSupplierContract}
    ID                  id-rosObject-respondingConsumerDSA }

indspContract CONTRACT ::= {
    CONNECTION          dspConnectionPackage
    INITIATOR CONSUMER OF { inChainedModifyPackage | inChainedSearchPackage |
                           chainedExecutePackage }
    ID                  id-contract-indsp }

dspConnectionPackage CONNECTION-PACKAGE ::= {
    BIND                dSABind
    UNBIND              in-DSAUnbind
    ID                  id-package-dspConnection }

in-DSAUnbind          OPERATION ::= inEmptyUnbind

inChainedModifyPackage OPERATION-PACKAGE ::= {
    CONSUMER INVOKES   { chainedAddEntry | chainedRemoveEntry | chainedModifyEntry }
    ID                  id-package-inChainedModify }

inChainedSearchPackage OPERATION-PACKAGE ::= {
    CONSUMER INVOKES   { chainedSearch }
    ID                  id-package-inChainedSearch }

chainedExecutePackage OPERATION-PACKAGE ::= {
    CONSUMER INVOKES   { chainedExecute }
    ID                  id-package-chainedExecute }

chainedExecute OPERATION ::= chained { execute }

```



```

bind                Bind {dSABind},
unbind              Unbind{in-DSAUnbind}}

inDirectoryShadowAbstractSyntax ABSTRACT-SYNTAX ::= {
    BasicDISP-PDUs
    IDENTIFIED BY    id-as-indirectoryShadowAS}

BasicDISP-PDUs ::= TCMMessage {{DISP-Invokable},{DISP-Returnable}}

DISP-Invokable OPERATION ::= {requestShadowUpdate | updateShadow | coordinateShadowUpdate}

DISP-Returnable OPERATION ::= {requestShadowUpdate | updateShadow | coordinateShadowUpdate}

inDirectoryDSAShadowBindingAbstractSyntax ABSTRACT-SYNTAX ::= {
    DISPBinding-PDUs
    IDENTIFIED BY    id-as-indsaShadowBindingAS}

DISPBinding-PDUs ::= CHOICE {
    bind                Bind {dSAShadowBind},
    unbind              Unbind {in-DSAShadowUnbind}}

inDirectorySystemAC APPLICATION-CONTEXT ::= {
    CONTRACT            indspContract
    DIALOGUE MODE      structured
    TERMINATION         basic
    ABSTRACT SYNTAXES  {dialogue-abstract-syntax |
                        inDirectorySystemAbstractSyntax |
                        inDirectoryDSABindingAbstractSyntax}
    APPLICATION CONTEXT NAME id-ac-indirectorySystemAC}

inDirectorySystemWith3seAC APPLICATION-CONTEXT ::= {
    CONTRACT            dspContract
    DIALOGUE MODE      structured
    TERMINATION         basic
    ADDITIONAL ASE     {id-se-threewayse}
    ABSTRACT SYNTAXES  {dialogue-abstract-syntax |
                        inDirectorySystemAbstractSyntax |
                        inDirectoryDSABindingAbstractSyntax |
                        inSESEAbstractSyntax}
    APPLICATION CONTEXT NAME id-ac-indirectorySystemWith3seAC}

inshadowSupplierInitiatedAC APPLICATION-CONTEXT ::= {
    CONTRACT            shadowSupplierContract

```

```

DIALOGUE MODE          structured
TERMINATION            basic
ABSTRACT SYNTAXES     {dialogue-abstract-syntax |
                       inDirectoryShadowAbstractSyntax |
                       inDirectoryDSAShadowBindingAbstractSyntax}
APPLICATION CONTEXT NAME id-ac-inShadowSupplierInitiatedAC}

inshadowSupplierInitiatedWith3seAC APPLICATION-CONTEXT ::= { #
    CONTRACT            shadowSupplierContract #
    DIALOGUE MODE       structured #
    TERMINATION         basic #
    ADDITIONAL ASE      {is-se-threewayse} #
    ABSTRACT SYNTAXES  {dialogue-abstract-syntax | #
                       inDirectoryShadowAbstractSyntax | #
                       inDirectoryDSAShadowBindingAbstractSyntax | #
                       inSESEAbstractSyntax } #
    APPLICATION CONTEXT NAME id-ac-inShadowSupplierInitiatedWith3seAC } #

inshadowConsumerInitiatedAC APPLICATION-CONTEXT ::= {
    CONTRACT            shadowConsumerContract
    DIALOGUE MODE       structured
    TERMINATION         basic
    ABSTRACT SYNTAXES  {dialogue-abstract-syntax |
                       inDirectoryShadowAbstractSyntax |
                       inDirectoryDSAShadowBindingAbstractSyntax}
    APPLICATION CONTEXT NAME id-ac-inShadowConsumerInitiatedAC}

inshadowConsumerInitiatedWith3seAC APPLICATION-CONTEXT ::= { #
    CONTRACT            shadowConsumerContract #
    DIALOGUE MODE       structured #
    TERMINATION         basic #
    ADDITIONAL ASE      {id-se-threewayse} #
    ABSTRACT SYNTAXES  {dialogue-abstract-syntax | #
                       inDirectoryShadowAbstractSyntax | #
                       inDirectoryDSAShadowBindingAbstractSyntax | #
                       inSESEAbstractSyntax } #
    APPLICATION CONTEXT NAME id-ac-inShadowConsumerInitiatedWith3seAC} #

END

9 . S C F - S C F インタフェース #
10 . S C F - C U S F インタフェース #
11 . S S F アプリケーションエンティティ手順 #

```


12 . S C Fアプリケーションエンティティ手順

12.1 概要

この節では、S C F - S S Fインタフェースに関するS C Fアプリケーションエンティティ(A E)手順の定義について記述する。この手順は、N o . 7信号方式(S S 7)の使用に基づく。他の信号方式もまた、利用可能である。

さらに他の能力がインプリメントに依存した方式で、S C P、A DあるいはS Nでサポートされるかもしれない。

I T U - T勧告Q . 7 0 0、T T C標準J T - Q 7 7 1及びI T U - T勧告Q . 1 4 0 0に定義されたアーキテクチャに従って、A Eはトランザクション機能応用部(T C)及び一つ以上のT Cユーザと呼ばれるA S Eを含む。以下の節では、T T C標準J T - Q 7 7 1で規定するプリミティブを用いてT Cとインタフェースする、T CユーザA S E及びS A C F / M A C F規則を定義する。

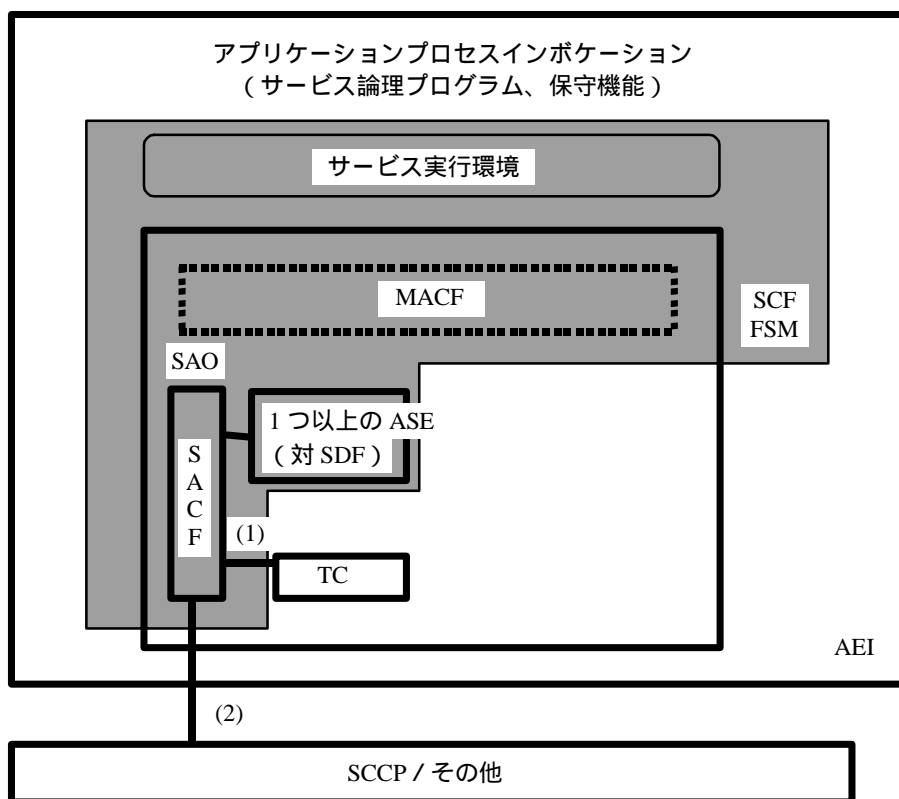
手順には、定義されたアプリケーションレイヤ構造をサポートしている他の信号メッセージ転送方式が同等に使用されるかもしれない。本文は決してサービス論理プログラム(S L P)への制限を指示するものではない。

本節で定義されるA E手順の解釈が、詳細手順やT Cサービスの利用規則と異なっている場合は、17章及び18章に含まれる記述や規則に従うこととする。

12.2 モデルとインタフェース

A E - S C Fの機能モデルを図12 - 1 / J T - Q 1 2 2 8に示す。A S EはS D Fと通信するためにサポートされたプロトコルレイヤにインタフェースし、サービス論理プログラム及び保守機能とインタフェースする。この標準の範囲は、図12 - 1 / J T - Q 1 2 2 8において影を付けた部分に限定される。

(注) S C F F S Mはいくつかの有限状態機構を含む。



- (1) TC プリミティブ
- (2) N プリミティブ

AEI: Application Entity Invocation (アプリケーションエンティティインボケーション)
 SCF: Service Control Function (サービス制御機能)
 FSM: Finite State Model (有限状態モデル)
 MACF: Multiple Association Control Function (複数アソシエーション制御機能)
 SACF: Single Association Control Function (単一アソシエーション制御機能)
 SAO: Single Association Object (単一アソシエーションオブジェクト)

注意: SCF FSM はいくつかの有限状態機構を含む
 注: 本標準では MACF については対象外である

図 12 - 1 / JT - Q 1 2 2 8 * SCF AE の機能モデル
 (ITU - T Q . 1 2 2 8)

図 12 - 1 / JT - Q 1 2 2 8 に示されたインタフェースは、TTC 標準 JT - Q 7 7 1 の中で規定された TC ユーザ ASE のプリミティブ (インタフェース (1)) を用いる。インテリジェントネットワークアプリケーションプロトコル (INAP) のオペレーションとパラメータは、この標準の第 8 章で定義されている。

12.3 SCF FSM と SLP / 保守機能間の相互関係

SCF FSM とサービス論理プログラム / 保守機能間のプリミティブインタフェースは、内部インタフェースであり、本標準での対象ではない。

SCSM インスタンス内には SDF 関連状態の FSM が存在する。

以下のテキストは、エンティティの機能的な能力よりむしろオペレーションの正確な順序を規定することを主要目的として、SCF と他の機能エンティティ間のインタフェースの手順の側面を系統的に記述している。従って、このテキストは SCF の機能的な能力のサブセットだけを記述している。

サ - ビス論理プログラムと S C F F S M間の相互関係は、(エンドユ - ザにより起動された呼と I Nサ - ビス論理により起動された呼の両方場合について、) 次のように記述されるかもしれない。

- I N呼処理要求が S S Fより受信された場合、S C F呼状態モデル (S C S M) のインスタンスが生成され、対応するサ - ビス論理プログラムが起動される。
- 呼の起動がサ - ビス論理から要求されると、S C S Mのインスタンスが生成される。

どちらの場合においても、S C F F S Mは要求に応じて S D F F S Mとの相互作用を処理し、必要に応じてイベントをサ - ビス論理プログラムに通知する。

S C Fから受信されたオペレ - ションの実行に関連した管理機能は、S C F管理エンティティ (S C M E) によって実行される。複数の要求が同時かつ非同期に S C Fにより起動され得る。この事は S C F F S Mオブジェクトの生成、起動および維持タスクを行う単一エントリが必要となることを意味する。

S C M EはS C M E制御
と複数のS C M E F S Mインスタンス #
により構成される。S C M E制御は、
それぞれ異なった複数の #
S C F呼状態モデル (S C S M) 及び機能エンティティアクセスマネ - ジャ (F E A M) とインタフェ - ス
する。図 1 2 - 2 / J T - Q 1 2 2 8はS C F F S M構造を示す。

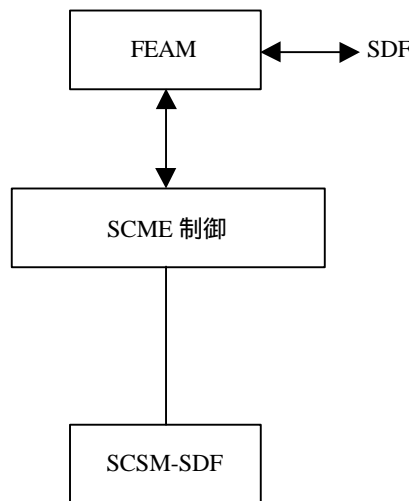


図 1 2 - 2 / J T - Q 1 2 2 8 * S C F F S M構造 #
(I T U - T Q . 1 2 2 8) *

SCSM-SDFはS D F F S Mとの相互作用を処理する。

S D Fから受信したオペレーションの実行に関連する管理機能はS C F管理エンティティ (S C M E) によって実行される。S C M EはS C M E制御
と複数のS C M E F S Mのインスタンス
により構成される。S C M E制御は機能エンティティアクセス管理 (F E A M) のみならず、
異なる #
S C S M
やS C M E F S Mそれぞれ #
とインタフェースする。

S C M E制御はS C F F S Mの全てのインスタンスに代わってS D Fとの関連を維持する。これらのS

CF FSMのインスタンスはSCFに関連するイベントの生起に従って同時にかつ非同期に生成される。この事はSCF FSMの生成、起動および維持タスクを行う単一エンティティが必要であることを意味する。特にSCME制御は以下のタスクを行う：

- (1) 別のFEからの入力メッセージを解釈し、それらに対応するSCSMイベントに翻訳する。
- (2) SCSM出力を別のFEへの対応するメッセージに翻訳する。
- (3) (呼処理と)非同期的な全ての動作を実行する。(そのような動作の一つはフロ-制御である。
- (4) SCFと他のFE間の持続的相互動作をサポートする。

最後に、機能エンティティアクセスマネ-ジャ(FEAM)はSCMEを低レベルのインタフェ-ス機能から解放する。FEAMの機能は次のようなものを含んでいる：

- (1) SDFとのインタフェ-スを設定し、維持すること。
- (2) SDFから受信されたメッセージをSCME制御に引き渡すこと(必要な時にはキューイングすること)。
- (3) SCME制御から受信したメッセージを編集し、(必要な時には)キューイング、SDFに送信すること。

12.4 部分的なSCF管理エンティティ(SCME)状態遷移図

#

12.5 SCF呼状態モデル(SCSM)

SCSM内にはSDF関連状態のFSMが存在する。

SCSMインスタンスはSCME制御から適当なFSM(SDF関連の)へ全てのイベントを送出する。FSMインスタンスが存在していない場合、SCME制御はインスタンスを生成し、そこにイベントを送出する。

一つ以上の状態に対し適用される一般的なルールは：

全ての状態において、受信したオペレーションにエラーがあった場合、SLP及び管理機能は通知される。

一般的にSCSMは同じ状態に留まるが、異なるエラー処理は16章に記述されるある特定のケースについて可能となる。オペレーションクラスにより、エラーはSDFに通知され得る。(TTC標準JT-Q774参照)

12.5.1 SSF/SRF関連状態(SCSM-SSF/SRF)

#

12.5.2 SDF関連状態(SCSM-SDF)

SDFとの相互動作は、SCFの任意の状態から可能である。以下の項ではSDF関連状態が規定されている。モデルは一つのSCFと一つのSDF間の関係を記述している。もし、SCFが他のSDFへアクセスする必要があるれば、新しいFSMが生成される。

以下では、状態とイベントに番号が付与されている。図12-18/JT-Q1228を参照のこと。

12.5.2.1 状態1: Idle

以下のイベントがこの状態において考慮される。

- ・(e1) Bind_Request: これは内部イベントであり、データへのアクセスを開始するためにSDFへの結合をサ-ビス論理が必要とすることにより生ずる。このイベントは状態2、Wait for subsequent requests への遷移を生ずる。

12.5.2.2 状態2: Wait for subsequent requests

この状態において、Bind オペレーションと共に(同一メッセージで)SDFへ送られるべき引き続くオペレーションが存在するかも知れない。以下2つのイベントがこの状態において考慮される。

- ・ (e2) Request_to_SDF : これは内部イベントであり、オペレーションを受信することにより生ずる。オペレーションはデリミタの受信 (またはタイマ満了) までバッファリングされる。SCSMは同一状態に留まる。
- ・ (e3) Request_to_SDF_with_Bind : これは内部イベントであり、デリミタの受信により引き起こされる。デリミタは送信されるべき最終オペレーションの受信を示している。いったんデリミタが受信されると、Bind オペレーションのアーギュメントと、もしあれば、その他のオペレーションのアーギュメントを含むメッセージがSDFへ送信される。このイベントは本状態から状態 3、Wait for Bind result への遷移を生じる。

12.5.2.3 状態 3 : Wait for Bind result

この状態において、SCFはSDFからの応答を待っている。2つのイベントがこの状態において考慮される。

- ・ (E4) Bind_Error : これは外部イベントであり、以前にSDFに対して発行された Bind オペレーションに対するエラー応答の受信により生じる。このイベントは本状態から状態 1、Idle への遷移を生ずる。
- ・ (E5) Response_from_SDF_with_Bind : これは外部イベントであり、以前にSDFに対して発行されたその他のオペレーション (もしあれば) と結合された Bind 結果の受信により生じる。このイベントは本状態から状態 4、SDF Bound への遷移を生ずる。

12.5.2.4 状態 4 : SDF Bound

この状態において、SCFはSDFとの間に認証済みアクセスを確立している。そして、サービス論理からSDFへの要求または、以前にSDFへ発行されたオペレーションに対する応答を待っている。3つのイベントがこの状態において考慮される。

- ・ (e6) Request_to_SDF : これは内部イベントであり、SDF内のデータへアクセスすることをサービス論理が必要とする場合に生ずる。SCSMは同一状態に留まる。
- ・ (E7) Response_from_SDF : これは外部イベントであり、以前にSDFに対して発行されたオペレーションに対する応答の受信により生ずる。SCSMは同一状態に留まる。
- ・ (e8) Unbind_Request : これは内部イベントであり、SDFとの間の認証済みアクセスをサービス論理が解放する必要が生じた時に生じる。このイベントは本状態から状態 1、Idle への遷移を生ずる。

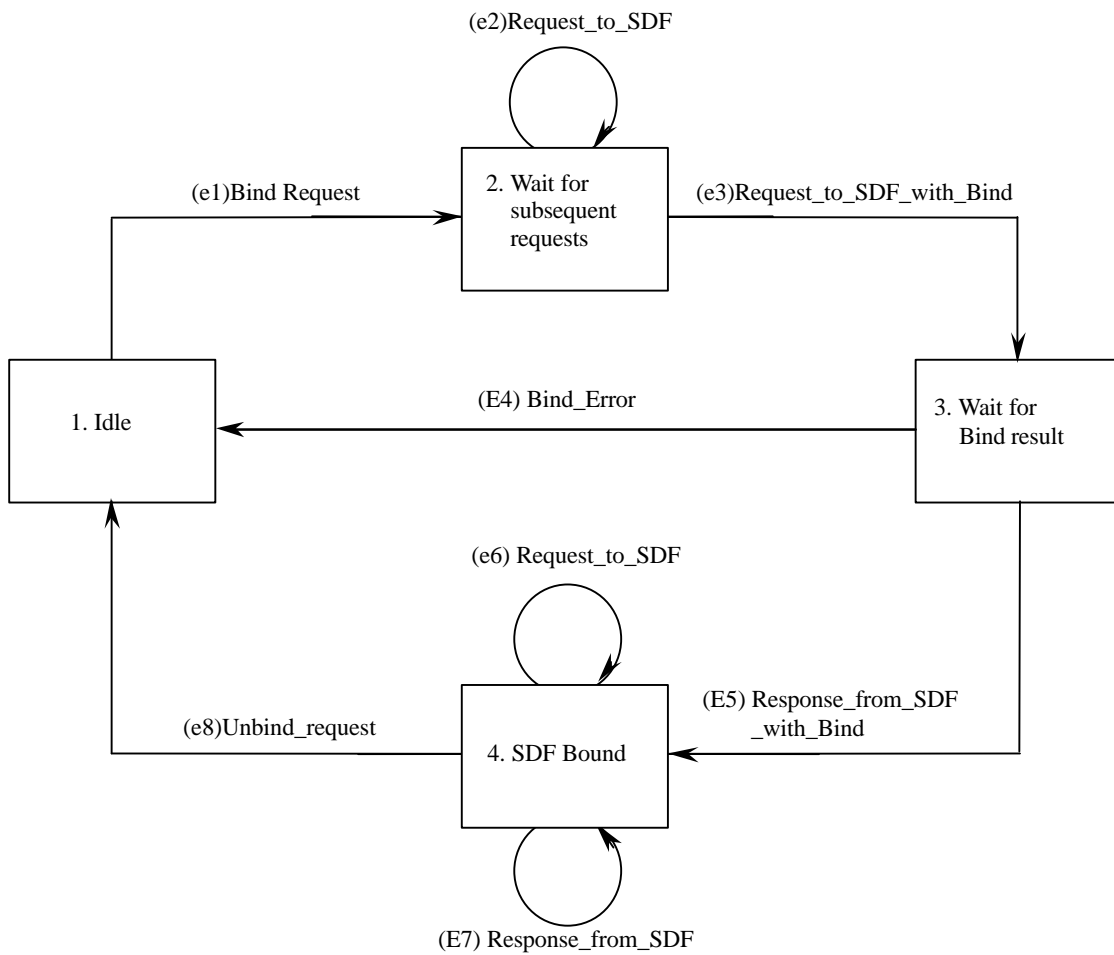


図 12 - 18 / JT - Q 1 2 2 8 SCSM FSM (SDF 関連部)
(ITU - T Q . 1 2 2 8)

- 12.5.3 SCF 関連状態 #
- 12.5.4 CUSF 関連状態 (SCSM-CUSF) #
- 12.5.5 USI_SCF FSM #
- 13 . SRF アプリケーション エンティティ 手順 #

14 . S D Fアプリケーションエンティティ手順

14.1 概要

この節では、S D F - S C FおよびS D F - S D Fインタフェースに関連したS D Fアプリケーションエンティティ (A E) 手順の定義について記述する。その手順はN o . 7 共通線信号方式 (S S 7) の利用を前提としている。

他の能力はS D P、A DもしくはS Nにおいてインプリメント依存な方法でサポートされるかもしれない。

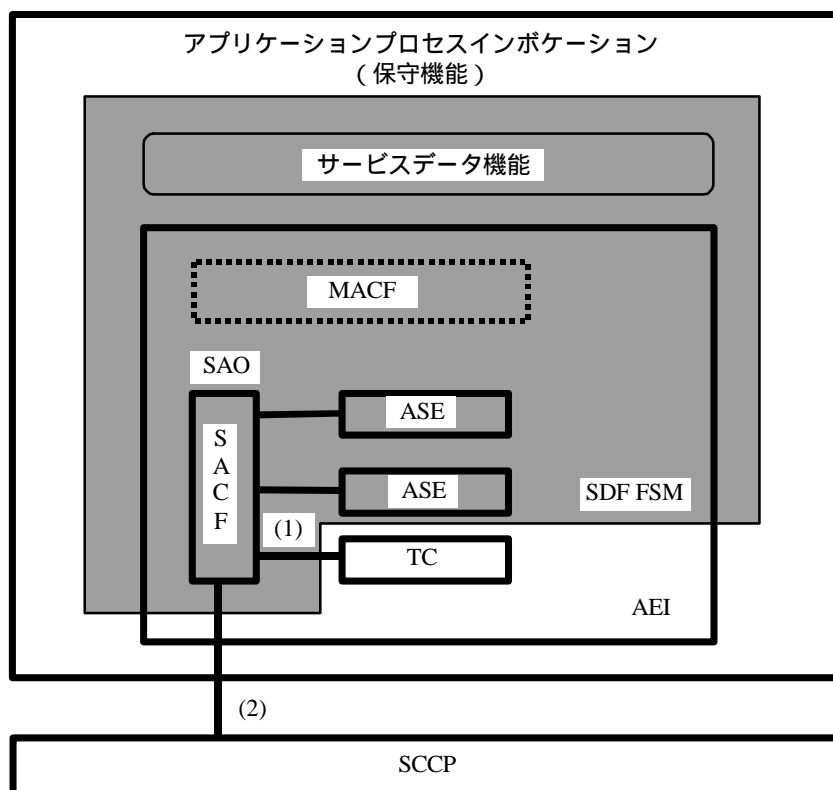
I T U - T 勧告Q . 7 0 0、T T C 標準J T - Q 7 7 1、およびI T U - T 勧告Q . 1 4 0 0 の中で定義されたアキテクチャに従って、A E は、トランザクション機能応用部 (T C) 及び一つ以上のT C ユーザと呼ばれるA S E を含み、これらはディレクトリに基づいて構成される (I T U - T 勧告X . 5 0 0 シリーズ)。以下の節は、T T C 標準J T - Q 7 7 1 で規定されたプリミティブを用いているT C とインタフェースする、T C ユーザA S E 及びS A C F / M A C F 規則を定義する。

その手順には、定義されたアプリケーションレイヤ構造をサポートしている、他の信号メッセージ転送方式で同等に用いられるかもしれない。

本章で定義されるA E 手順が 17 章と異なる場合には、17 章で定義されるルールに従うこととする。

14.2 モデルとインタフェース

A E - S D F の機能モデルを図 1 4 - 1 / J T - Q 1 2 2 8 に示す。A S E はS C F および他のS D F と通信するためにT C とインタフェースし、保守機能とインタフェースする。この標準の範囲は、図 1 4 - 1 / J T - Q 1 2 2 8 において影をつけた部分に限定される。



- (1) TC プリミティブ
- (2) N プリミティブ

AEI: Application Entity Invocation (アプリケーションエンティティインボケーション)
 SDF: Service Data Function (サービスデータ機能)
 FSM: Finite Status Model (有限状態モデル)
 SACF: Single Association Control Function (単一アソシエーション制御機能)
 SAO: Single Association Object (単一アソシエーションオブジェクト)

注意: SDF FSM はいくつかの有限状態機構を含む

図 14 - 1 / JT - Q 1 2 2 8 SDF の機能モデル
 (ITU - T Q . 1 2 2 8)

図 14 - 1 / JT - Q 1 2 2 8 に示されたインタフェースは、TTC 標準 JT - Q 7 7 1 の中で規定された TC ユーザ ASE のプリミティブ (インタフェース (1)) を用いる。網間インテリジェントネットワークアプリケーションプロトコル (INAP) のオペレーションとパラメータは、この標準の 4 章から 10 章 8 章 で定義されている。

*

以下のいずれかの事象が生じた際に、SDF FSM のインスタンスが生成される。

- (1) IN の呼に関するあるいは呼と関連しない処理を SCF から受信した場合
- (2) SDF が別の SDF に連鎖したオペレーションを発行あるいは別の SDF から連鎖したオペレーションを受信した場合
- (3) SDF が別の SDF にシャドウイングオペレーションを発行あるいは別の SDF からシャドウイングオペレーションを受信した場合

SDF FSM は、SCF FSM および別の SDF FSM との相互作用を処理する。

14.3 SDF FSM構造

SDF FSMの構成を図14-2/JT-Q1228に示す。

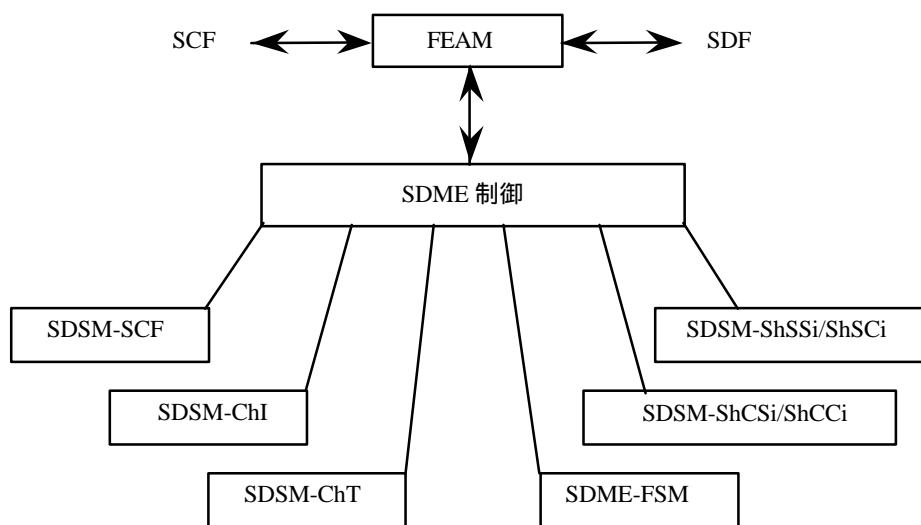


図14-2/JT-Q1228 SDFインタフェース
(ITU-T Q.1228)

SDF FSM (SDSM-SCF) はSCF FSMとの相互作用を処理する。SDSM-ChI および SDSM-ChT は連鎖の起動および停止に関するSDF間の相互作用を処理する。SDSM-ShSSI および SDSM-ShCSI はシャドウイングの供給側起動による供給側と消費側の為のSDF間の相互作用を処理する。SDSM-ShCSI および SDSM-ShCCi は消費側起動による上記相互作用を処理する。SDME FSMはSDFとSDF管理機能間の相互作用を処理する。

SCFあるいは協調しているSDFから受信したオペレーションの実行に関する管理機能は、SDF管理エンティティ(SDME)により実行される。SDMEはSDME制御およびSDME FSMの複数のインスタスからなる。SDME制御は、機能エンティティアクセスマネージャ(FEAM)のみならず、異なるSDF FSM(例、SDSM-SCF)およびSDME FSMともそれぞれインタフェースする。

FEAMは以下を含む下位レベルのインタフェース保守機能を提供する。

- (1) SCFおよび協調しているSDFとのインタフェースを設定し、維持すること。
- (2) SCFおよび協調しているSDFから受信したメッセージのSDME制御に引き渡すこと。(必要なときはキューイングすること)
- (3) SDME制御から受信したメッセージの編集、(必要であれば)キューイング、およびSCFおよび協調しているSDFへ送信すること。

SDME制御は、すべてのSDF FSM(例、SDSM-SCF、SDSM-ChI)のインスタスに代わって、SCFおよび協調しているSDFとのアソシエーションを保守する。SDF FSMのこれらのインスタスは、SDFに関連するイベントの生起にしたがって同時にあるいは非同期に生起する。これはすなわち、SDF FSMの生成、起動および保守を実行するためには唯一のエンティティが必要であることを意味する。特に、SDME制御は以下の機能を実行する。

- (1) 他のFEからの入力メッセージを解釈し、対応するSDF FSMイベントに変換する。
- (2) SDF FSMからの出力を他のFEへの対応するメッセージに変換する。
- (3) SDF内の管理および監視機能に関する非同期の動作を検出し、SDME FSMのインスタスを

生成する。例えば、網事業者間でのシャドウイング手順の実施管理である。この場合、SDME制御は本管理に関するオペレーションを処理するためにSDME FSMのインスタンスを生成する。

14.4 SDF 状態遷移モデル

14.4.1 SCF 関連状態に対する SDF 状態遷移モデル

網間INAPにおけるSCFとの相互作用に関する、SDFのジョブ、Bind手順後のSCFからの全ての要求に（同期して）応答することである。図14-3/JT-Q1228に有限状態モデル(FSM)を示す。

各状態は以下の各項で論じられている。

一つ以上の状態に適用可能な一般的な規則は以下のようである。

任意の状態において、SCFとの間の対話が終了されるとSDF FSMは、呼に割り当てられたどのリソ-スも解放されていることを保証した後で、Idle 状態に戻る。

14.4.1.1 状態 1 : Idle

この状態において許容されるただ一つのイベントは：

- ・ (E1) Bind_From_SCF：これは外部イベントであり、SCFからのDirectory Bind オペレーションの受信により生ずる。このイベントは本状態から状態 2、Bind Pending への遷移を生ずる。

14.4.1.2 状態 2 : Bind Pending

この状態においては、結合要求がSCFから受信されている。受信したSDFは結合の設定の動作と同時にSCFへのアクセスコントロールを実行する（例えば、アクセスに関する認証）。しかし、ディレクトリ結合(DirectoryBind)オペレーションがダミーということもあり得る。この時、アクセスに関する認証は不要である。本状態では3つのイベントが生じ得る：

- ・ (e2) Bind_Error：これは内部イベントであり、以前にSDFに対して発行されたDirectoryBindオペレーションが失敗したことにより生じる。このイベントは状態 1、Idle への遷移を生ずる。DirectoryBindオペレーションに対するエラーが起動元SCFへ返される。
- ・ (E3) Request_from_SCF：これは外部イベントであり、DirectoryBindオペレーションの結果が決定する前にオペレーションを受信することにより生じる。

それは次のオペレーションの一つを伴う。

- ・ search
- ・ addEntry
- ・ removeEntry
- ・ modifyEntry
- ・ execute

オペレーションは記憶され、そしてSDSMは同一状態に留まる。別の状態に遷移が起るとき、そのオペレーションはそれらがその状態に起こったかのように再び見られる。

- ・ (e4) Bind_Successful：これは内部イベントであり、以前にSDFに対して発行されたDirectoryBindオペレーションが正常に完了したことにより生じる。このイベントは状態 3、SCF Bound への遷移を生ずる。

14.4.1.3 状態 3 : SCF Bound

この状態において、SDFに対するSCFのアクセスは認証されており、SCFからのオペレーションが受け付けられている。SCFからの要求待ちの他に、この状態でSDFは以前に発行されたオペレーション

への応答を送信することができる。本状態では3つのイベントが生じ得る：

- ・ (E5) Unbind_from_SCF：これは外部イベントであり、SCFからDirectory Unbind オペレーションを受信することにより生じる。SCF - SDF間アソシエーションは終了され、全ての関連するリソースは解放される。このイベントは本状態から状態1、Idleへの遷移を生ずる。
- ・ (e6) Response_to_SCF：これは内部イベントであり、SCFから発行されたオペレーションの実行が完了する、あるいはSCFに対するリフェラルエラーを生成することにより生じる。この時、応答またはリフェラルがSCFへ返される。SDSMは同一状態に留まる。
- ・ (E7) Request_from_SCF：これは外部イベントであり、SCFからSDFに対しての要求を受信することにより生じる。SDSMは同一状態に留まる。

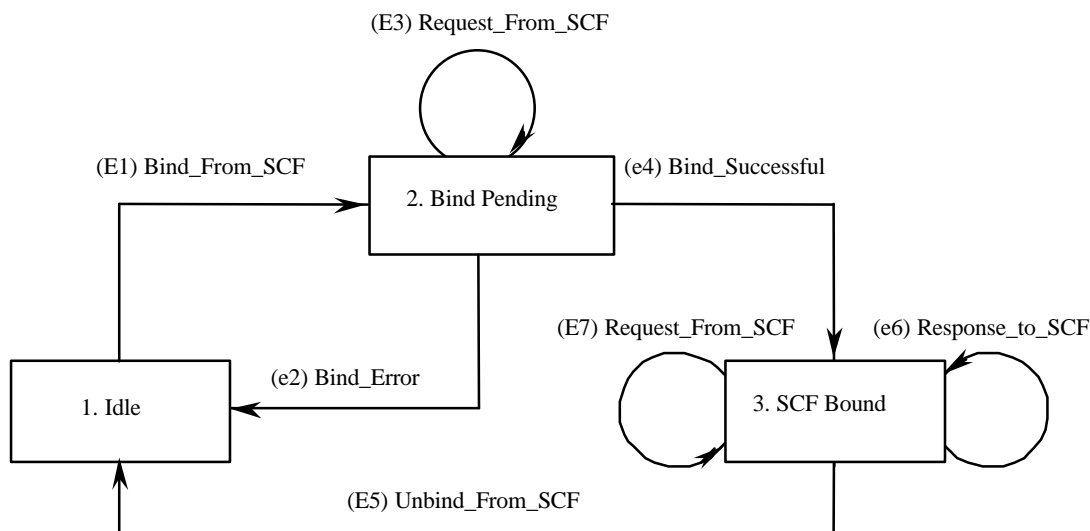


図14-3 / JT-Q1228 SDF FSM
(ITU-T Q.1228)

14.4.2 SDF関連状態に対するSDF状態遷移モデル

他のSDFとの相互作用に関しては、SDFはシャドウイングおよび連鎖オペレーションを実行する。シャドウイングのためのSDF / SDF相互作用に関する状態を14.4.2.1項に示す。また、連鎖のためのSDF / SDF相互作用に関する状態を14.4.2.2項に示す。

14.4.2.1 シャドウイングに対するSDF状態遷移モデル

シャドウイング手順に関しては、SDFはコピー供給側にもコピー消費側にもなり得る。さらに、シャドウイング手順はコピー供給側だけでなく、コピー消費側からも起動され得る。したがって、以下に示すように全部で4種類のFSMが存在する。

4つのSDF FSMをまとめて1つの複雑なFSMを定義することもできるが、SDFの異なる役割を明確に示すため、4つのFSMを別々に記述する方が良い。

以下のFSMでは、DSAShadowBindオペレーションと他のDISPオペレーションを一つのTCメッセージと一緒に送信する可能性を考慮している。

【シャドウ供給側起動による供給側状態機構 (SDSM-ShSSi)】

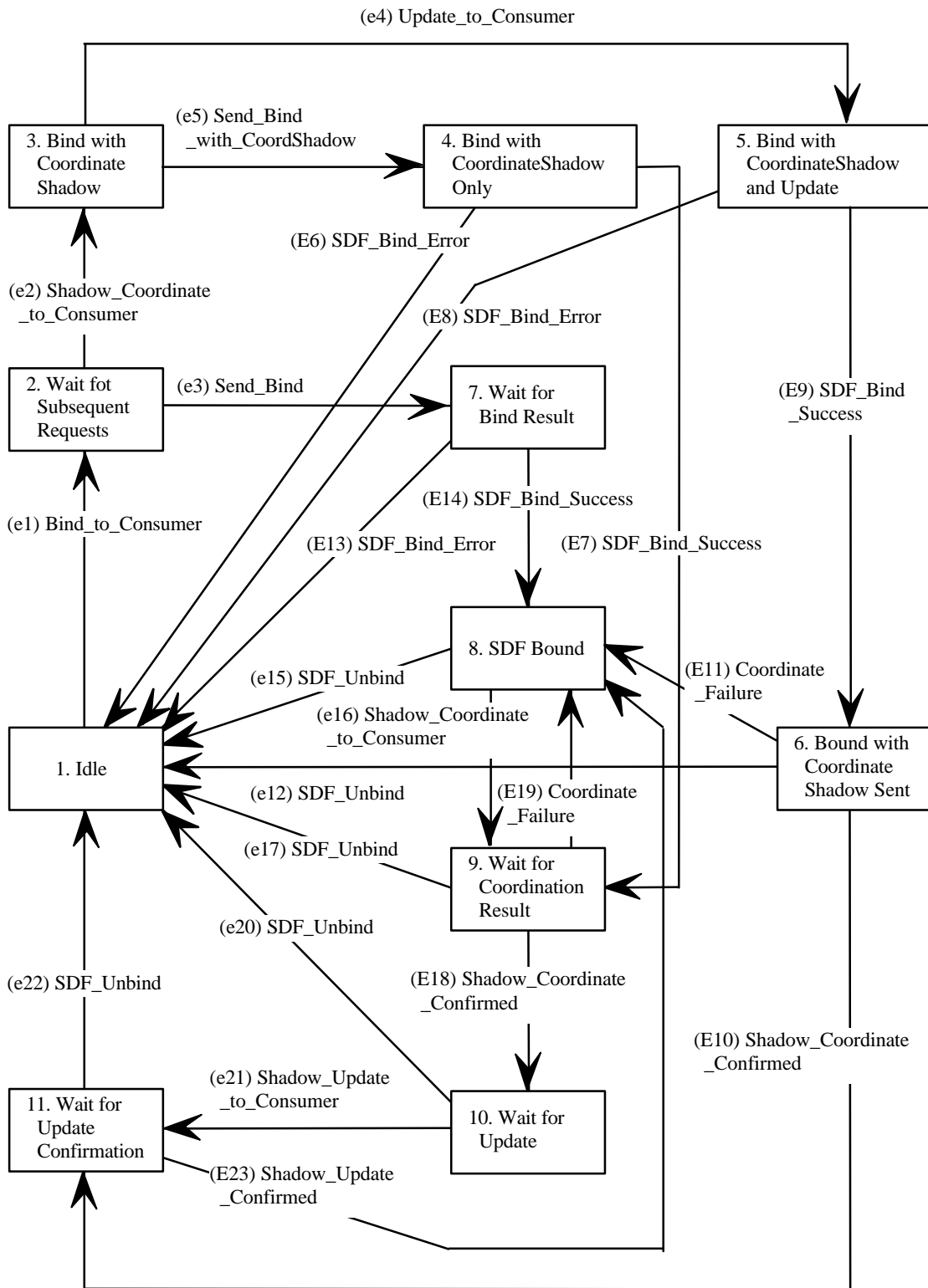


図14-4 / JT-Q1228 シャドウ供給側起動による供給側状態機構 (SDSM-ShSSi) (ITU-T Q.1228)

状態1 : Idle

この状態で許容される唯一のイベントは以下である。

- ・ (e1) Bind_to_Consumer : これは内部イベントであり、DSAShadowBind オペレーションを送信することにより生じる。これにより状態 2、Wait for Subsequent Requests に遷移する。

状態 2 : Wait for Subsequent Requests

この状態では、DSAShadowBind オペレーションと同一メッセージで消費側に送信する CoordinateShadowUpdate オペレーションを待っている。以下の 2 つのイベントが考えられる。

- ・ (e2) Shadow_Coordinate_to_Consumer : これは内部イベントであり、CoordinateShadowUpdate オペレーションを受信することにより生じる。オペレーションはデリミタ (あるいはタイマ満了) を受信するまでバッファリングされる。これにより状態 3、Bind with Coordinate Shadow に遷移する。
- ・ (e3) Send_Bind : これは内部イベントであり、送信すべき最終のオペレーションを受信あるいはタイマが満了したことを示すデリミタを受信することにより生じる。この内部イベントが受信されると、DSAShadowBind オペレーションを含む TC メッセージが消費側 SDF に対して送信される。これにより状態 7、Wait for Bind Results に遷移する。

状態 3 : Bind with Coordinate Shadow

この状態では、DSAShadowBind および CoordinateShadowUpdate の各オペレーションと共に送信される UpdateShadow オペレーション、あるいはデリミタを待っている。以下の 2 つのイベントが考えられる。

- ・ (e4) Update_to_Consumer : これは内部イベントであり、UpdateShadow オペレーションを受信することにより生じる。これにより DSAShadowBind、CoordinateShadowUpdate、および UpdateShadow オペレーションを含む TC メッセージが消費側 SDF へ送信される。これにより状態 5、Bind with Coordinate Shadow and Update に遷移する。
- ・ (e5) Send_Bind_with_CoordShadow : これは内部イベントであり、送信すべき最後の TC メッセージを受信したことを表すデリミタを受信するか、あるいはタイマが満了することにより生じる。この内部イベントが受信されると、DSAShadowBind および CoordinateShadowUpdate オペレーションを含む TC メッセージが消費側 SDF へ送信される。これにより状態 4、Bind with Coordinate Shadow Only に遷移する。

状態 4 : Bind with Coordinate Shadow Only

この状態では、消費側 SDF からの DSAShadowBind オペレーションの結果を待っている。以下の 2 つのイベントが考えられる。

- ・ (E6) SDF_Bind_Error : これは外部イベントであり、消費側に発行した DSAShadowBind オペレーションが失敗したことにより生じる。DSAShadowBind エラーがすでに返送されている。これにより状態 1、Idle に遷移する。
- ・ (E7) SDF_Bind_Success : これは外部イベントであり、DSAShadowBind オペレーションの結果を受信したことにより生じる。これは消費側に発行した DSAShadowBind オペレーションが成功したことを表している。これにより状態 9 : Wait for Coordination Result に遷移する。

状態 5 : Bind with Coordinate Shadow and Update

この状態では、消費側 SDF からの DSAShadowBind オペレーションの結果を待っている。以下の 2 つのイベントが考えられる。

- ・ (E8) SDF_Bind_Error : これは外部イベントであり、消費側に発行した DSAShadowBind オペレーションが失敗したことにより生じる。DSAShadowBind エラーがすでに返送されている。これにより状

態 1、Idle に遷移する。

- ・ (E9) SDF_Bind_Success : これは外部イベントであり、DSAShadowBind オペレーションの結果を受信したことにより生じる。これは消費側に発行した DSAShadowBind オペレーションが成功したことを表している。これにより状態 6、Bound with Coordinate Shadow Sent に遷移する。

状態 6 : Bound with Coordinate Shadow Sent

この状態では、消費側 S D F からの CoordinateShadowUpdate オペレーションの結果を待っている。以下の 3 つのイベントが考えられる。

- ・ (E10) Shadow_Coordinate_Confirmed : これは外部イベントであり、CoordinateShadowUpdate オペレーションに対する結果を受信することにより生じる。これは消費側 S D F に発行した CoordinateShadowUpdate オペレーションが成功したことを表している。これにより状態 11、Wait for Update Confirmation に遷移する。
- ・ (E11) Coordinate_Failure : これは外部イベントであり、発行した CoordinateShadowUpdate オペレーションに対するエラーを受信することにより生じる。これにより状態 8、SDF Bound に遷移する。
- ・ (e12) SDF_Unbind : これは内部イベントであり、2 つの S D F 間に確立した “ 認証されたアソシエーション ” を終了する必要性により生じる (例、ユーザの解放手順)。これにより状態 1、Idle に遷移する。

状態 7 : Wait for Bind Result

この状態では、シャドウ消費側からの DSAShadowBind オペレーションの結果を待っている。以下の 2 つのイベントが考えられる。

- ・ (E13) SDF_Bind_Error : これは外部イベントであり、消費側 S D F に発行した DSAShadowBind オペレーションが失敗したことにより生じる。DSAShadowBind エラーがすでに返送されている。これにより状態 1、Idle に遷移する。
- ・ (E14) SDF_Bind_Success : これは外部イベントであり、消費側 S D F に発行した DSAShadowBind オペレーションが成功したことにより生じる。これにより状態 8、SDF Bound に遷移する。

状態 8 : SDF Bound

この状態では、S D F は消費側への “ 認証されたアソシエーション ” を確立し、CoordinateShadowUpdate オペレーションを送信する準備が完了している。以下の 2 つのイベントが考えられる。

- ・ (e15) SDF_Unbind : これは内部イベントであり、2 つの S D F 間に確立した “ 認証されたアソシエーション ” を終了する必要性による (例、ユーザの解放手順) か、あるいは in-DSAShadowUnbind オペレーションを発行するために生じる。これにより状態 1、Idle に遷移する。
- ・ (e16) Shadow_Coordinate_to_Consumer : これは内部イベントであり、消費側 S D F に対して後に更新するためにシャドウ更新の調整の要求を送信するために生じる。これにより状態 9、Wait for Coordination Result に遷移する。

状態 9 : Wait for Coordination Result

この状態では、供給側 S D F が CoordinateShadowUpdate 要求を送信し、消費側 S D F からの結果を待っている。以下の 3 つのイベントが考えられる。

- ・ (e17) SDF_Unbind : これは内部イベントであり、2 つの S D F 間に確立した “ 認証されたアソシエーション ” を終了する必要性により生じる (例、ユーザの解放手順)。これにより状態 1、Idle に遷移する。

- ・ (E18) Shadow_Coordinate_Confirmed : これは外部イベントであり、発行した CoordinateShadowUpdate オペレーションに対する応答を受信することにより生じる。これにより状態 10、Wait for Update に遷移する。
- ・ (E19) Coordinate_Failure : これは外部イベントであり、発行した CoordinateShadowUpdate オペレーションに対するエラーを受信することにより生じる。これにより状態 8、SDF Bound に遷移する。

状態 10 : Wait for Update

この状態では、供給側 S D F が、発行した CoordinateShadowUpdate 要求に対する確認を受信し、消費側に UpdateShadow 要求を送信する準備が完了している。以下の 2 つのイベントが考えられる。

- ・ (e20) SDF_Unbind : これは内部イベントであり、2 つの S D F 間に確立した “ 認証されたアソシエーション ” を終了する必要性により生じる (例、ユーザの解放手順)。これにより状態 1、Idle に遷移する。
- ・ (e21) Shadow_Update_to_Consumer : これは内部イベントであり、シャドウの更新要求を消費側 S D F へ送信するために生じる。これにより状態 11、Wait for Update Confirmation に遷移する。

状態 11 : Wait for Update Confirmation

この状態では、S D F は UpdateShadow 要求を送信し、消費側からの結果を待っている。以下の 2 つのイベントが考えられる。

- ・ (e22) SDF_Unbind : これは内部イベントであり、2 つの S D F 間に確立した “ 認証されたアソシエーション ” を終了する必要性により生じる (例、ユーザの解放手順)。これにより状態 1、Idle に遷移する。
- ・ (E23) Shadow_Update_Confirmed : これは外部イベントであり、発行した UpdateShadow オペレーションに対する応答を受信することにより生じる。これにより状態 8、SDF Bound に遷移する。

【シャドウ消費側起動による供給側状態機構 (SDSM-ShSCi)】

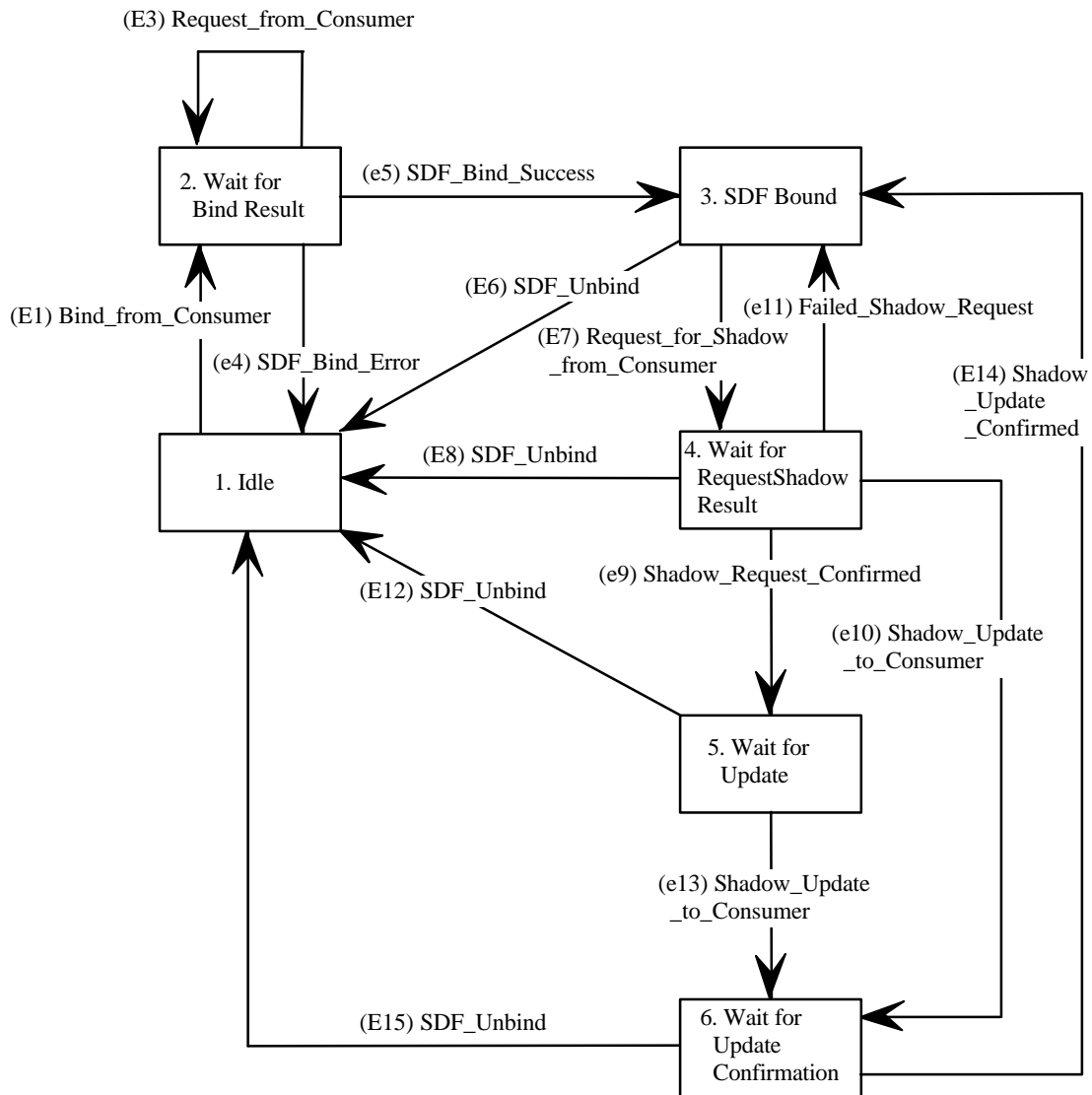


図 1 4 - 5 / J T - Q 1 2 2 8 シャドウ消費側起動による供給側状態機構 (SDSM-ShSCi)
(I T U - T Q . 1 2 2 8)

状態 1 : Idle

この状態で許容される唯一のイベントは以下である。

- ・ (E1) Bind_from_Consumer : これは外部イベントであり、 DSAShadowBind オペレーションを受信することにより生じる。これにより状態 2、 Wait for Bind Result に遷移する。

状態 2 : Wait for Bind Result

この状態では、 DSAShadowBind オペレーションを実行している。以下の 3 つのイベントが考えられる。

- ・ (E3) Request_from_Consumer : これは外部イベントであり、 DSAShadowBind オペレーションの結果が判明する前に RequestShadowUpdate オペレーションを受信することにより生じる。これは RequestShadowUpdate オペレーションが結合要求と同一の TC メッセージにより送信された時に発

生ずる。RequestShadowUpdate メッセージは蓄積され、同一の状態に遷移する。次の状態に遷移する時に、RequestShadowUpdate オペレーションは遷移元の状態において発生したと見なし再検査される。

- ・ (e4) SDF_Bind_Error : これは内部イベントであり、消費側から発行された DSAShadowBind オペレーションが失敗したことにより生じる。DSAShadowBind エラーが返送される。これにより状態 1、Idle に遷移する。
- ・ (e5) SDF_Bind_Success : これは内部イベントであり、消費側から発行された DSAShadowBind オペレーションが成功したことにより生じる。これにより DSAShadowBind オペレーションの結果が返送され、状態 3、SDF Bound に遷移する。

状態 3 : SDF Bound

この状態では、供給側 S D F は消費側 S D F からの RequestShadowUpdate オペレーションを待っている。以下の 2 つのイベントが考えられる。

- ・ (E6) SDF_Unbind : これは外部イベントであり、2 つの S D F 間に確立した “ 認証されたアソシエーション ” を終了する必要性による (例、ユーザの解放手順) か、あるいは in-DSAShadowUnbind オペレーションを受信することにより生じる。これにより状態 1、Idle に遷移する。
- ・ (E7) Request_for_Shadow_from_Consumer : これは外部イベントであり、消費側 S D F からの RequestShadowUpdate オペレーションを受信することにより生じる。これにより状態 4、Wait for RequestShadow Result に遷移する。

状態 4 : Wait for RequestShadow Result

この状態では、供給側 S D F は消費側 S D F からの RequestShadowUpdate オペレーションを受信している。以下の 4 つのイベントが考えられる。

- ・ (E8) SDF_Unbind : これは外部イベントであり、2 つの S D F 間に確立した認証されたアソシエーションを終了することにより生じる (例、ユーザの解放手順)。これにより状態 1、Idle に遷移する。
- ・ (e9) Shadow_Request_Confirmed : これは内部イベントであり、更新の合意が受け入れられることを表している。これは供給側 S D F が受信した RequestShadowUpdate オペレーションに対する応答を送信することにより生じる。これにより状態 5、Wait for Update に遷移する。
- ・ (e10) Shadow_Update_to_Consumer : これは内部イベントであり、更新の合意が受け入れられかつ、UpdateShadow メッセージの送信が準備されていることを表している。これは同一の T C メッセージで、RequestShadowUpdate の結果および UpdateShadow オペレーションを送ることにより生じる。これにより状態 6、Wait for Update Confirmation に遷移する。
- ・ (e11) Failed_Shadow_Request : これは内部イベントであり、受信した RequestShadowUpdate オペレーションに対するエラーを送信することにより生じる。これにより状態 3、SDF Bound に遷移する。

状態 5 : Wait for Update

この状態では、供給側 S D F がすでに受信した RequestShadowUpdate オペレーションに対する応答を送信し、消費側 S D F に対して UpdateShadow オペレーションを送信する準備が完了している。以下の 2 つのイベントが考えられる。

- ・ (E12) SDF_Unbind : これは外部イベントであり、2 つの S D F 間に確立した認証されたアソシエーションを終了する必要性により生じる (例、ユーザの解放手順)。これにより状態 1、Idle に遷移する。
- ・ (e13) Shadow_Update_to_Consumer : これは内部イベントであり、シャドウ更新の要求を消費側 S D F へ送信することにより生じる。これにより状態 6、Wait for Update Confirmation に遷移する。

状態 6 : Wait for Update Confirmation

この状態では、供給側 S D F は UpdateShadow 要求を送信し、消費側 S D F からの応答を待っている。以下の 2 つのイベントが考えられる。

- ・ (E14) Shadow_Update_Confirmed : これは外部イベントであり、発行した UpdateShadow オペレーションに対する応答を受信することにより生じる。これにより状態 3、SDF Bound に遷移する。
- ・ (E15) SDF_Unbind : これは外部イベントであり、S D F 間に確立した認証されたアソシエーションを終了する必要性により生じる (例、ユーザの解放手順)。これにより状態 1、Idle に遷移する。

【シャドウ供給側起動による消費側状態機構 (SDSM-ShCSI)】

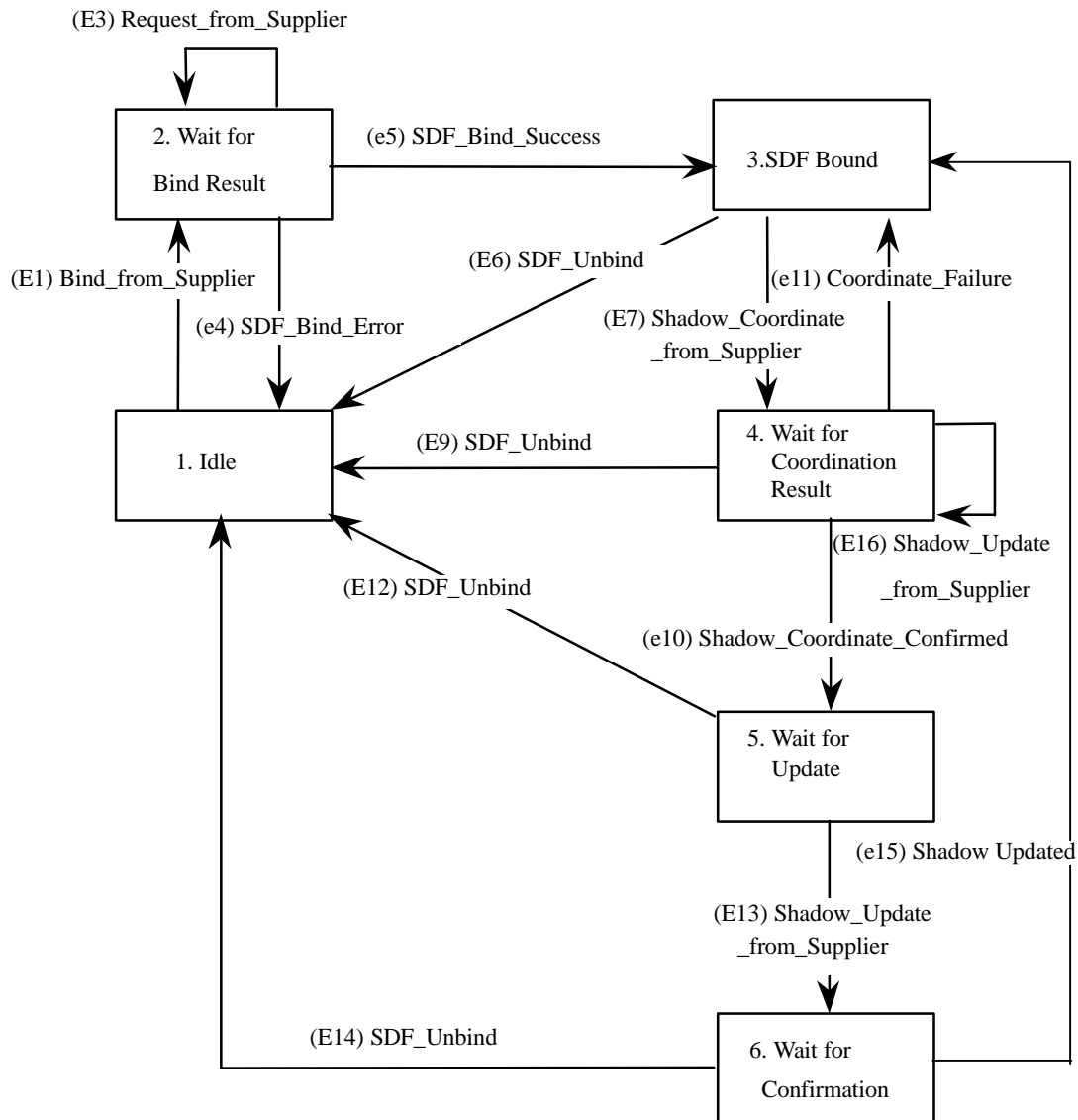


図 1 4 - 6 / J T - Q 1 2 2 8 シャドウ供給側起動による消費側状態機構 (SDSM-ShCSI)
(I T U - T Q . 1 2 2 8)

状態 1 : Idle

この状態で許容される唯一のイベントは以下である。

- ・ (E 1) Bind_from_Supplier : これは外部イベントであり、DSAShadowBind オペレーションを受信することにより生じる。これにより状態 2、Wait for Bind Result に遷移する。

状態 2 : Wait for Bind Result

この状態では、消費側 S D F は DSAShadowBind オペレーションを受信し、応答しようとしている。以下の 3 つのイベントが考えられる。

- ・ (E 3) Request_from_Supplier : これは外部イベントであり、DSAShadowBind オペレーションの結果が判明する前にオペレーションを受信することにより生じる。これは CoordinateShadowUpdate または UpdateShadow が、結合要求と同じ T C メッセージで送られた時に生じる。これらのオペレーション

は受け入れられ、同一の状態に留まる。次の状態への遷移が起きたときに、これらのオペレーションはその状態で受信されたとみなされ再確認される。

- ・ (e4) SDF_Bind_Error : これは内部イベントであり、供給側 S D F から発行された DSAShadowBind オペレーションが失敗したことにより生じる。DSAShadowBind エラーが返送される。これにより状態 1、Idle に遷移する。
- ・ (e5) SDF_Bind_Success : これは内部イベントであり、供給側 S D F から発行された DSAShadowBind オペレーションが成功したことにより生じる。これにより状態 3、SDF Bound に遷移する。

状態 3 : SDF Bound

この状態では、消費側 S D F は供給側 S D F からの CoordinateShadowUpdate オペレーションを待っている。以下の 2 つのイベントが考えられる。

- ・ (E6) SDF_Unbind : これは外部イベントであり、2 つの S D F 間に確立した認証されたアソシエーションを終了 (例、ユーザの解放手順) あるいは in-DSAShadowUnbind オペレーションの受信により生じる。これにより状態 1、Idle に遷移する。
- ・ (E7) Shadow_Coordinate_from_Supplier : これは外部イベントであり、供給側 S D F からの CoordinateShadowUpdate オペレーションを受信することにより生じる。これにより状態 4、Wait for Coordination Result に遷移する。

状態 4 : Wait for Coordination Result

この状態では、消費側 S D F が供給側 S D F からの CoordinateShadowUpdate オペレーションを受信し、それを処理している。以下の 4 つのイベントが考えられる。

- ・ (E9) SDF_Unbind : これは外部イベントであり、2 つの S D F 間に確立した認証されたアソシエーションを終了することにより生じる (例、ユーザの解放手順)。これにより状態 1、Idle に遷移する。
- ・ (e10) Shadow_Coordinate_Confirmed : これは内部イベントであり、供給側 S D F から受信した CoordinateShadowUpdate オペレーションが成功したことにより生じる。これにより状態 5、Wait for Update に遷移する。
- ・ (e11) Coordinate_Failure : これは内部イベントであり、受信した CoordinateShadowUpdate オペレーションに対するエラーを送信することにより生じる。これにより状態 3、SDF Bound に遷移する。
- ・ (E16) Shadow_Update_from_Supplier : これは外部イベントであり、これは CoordinateShadowBind と CoordinateShadowUpdate と同じ TC メッセージにより UpdateShadow オペレーションを受信することにより生じる。UpdateShadow メッセージは受け入れられ同一の状態に留まる。次の状態に遷移するときに UpdateShadow はその状態で受信されたとみなされ再確認される。

状態 5 : Wait for Update

この状態では、消費側 S D F は UpdateShadow オペレーションを待っている。以下の 2 つのイベントが考えられる。

- ・ (E12) SDF_Unbind : これは外部イベントであり、2 つの S D F 間に確立した認証されたアソシエーションを終了することにより生じる (例、ユーザの解放手順)。これにより状態 1、Idle に遷移する。
- ・ (E13) Shadow_Update_from_Supplier : これは外部イベントであり、供給側 S D F からの UpdateShadow オペレーションを受信することにより生じる。これにより状態 6、Wait for Update Confirmation に遷移する。

状態 6 : Wait for Update Confirmation

この状態では、消費側 S D F が供給側 S D F からの UpdateShadow オペレーションを受信し、コピーの更新を処理している。以下の 2 つのイベントが考えられる。

- ・ (E14) SDF_Unbind : これは外部イベントであり、2 つの S D F 間に確立した認証されたアソシエーションを終了することにより生じる (例、ユーザの解放手順)。これにより状態 1、Idle に遷移する。
- ・ (e15) Shadow_Updated : これは内部イベントであり、UpdateShadow オペレーションを正常に完了し、応答を送信することにより生じる。これにより状態 3、SDF Bound に遷移する。

【シャドウ消費側起動による消費側状態機構 (SDSM-ShCCi)】

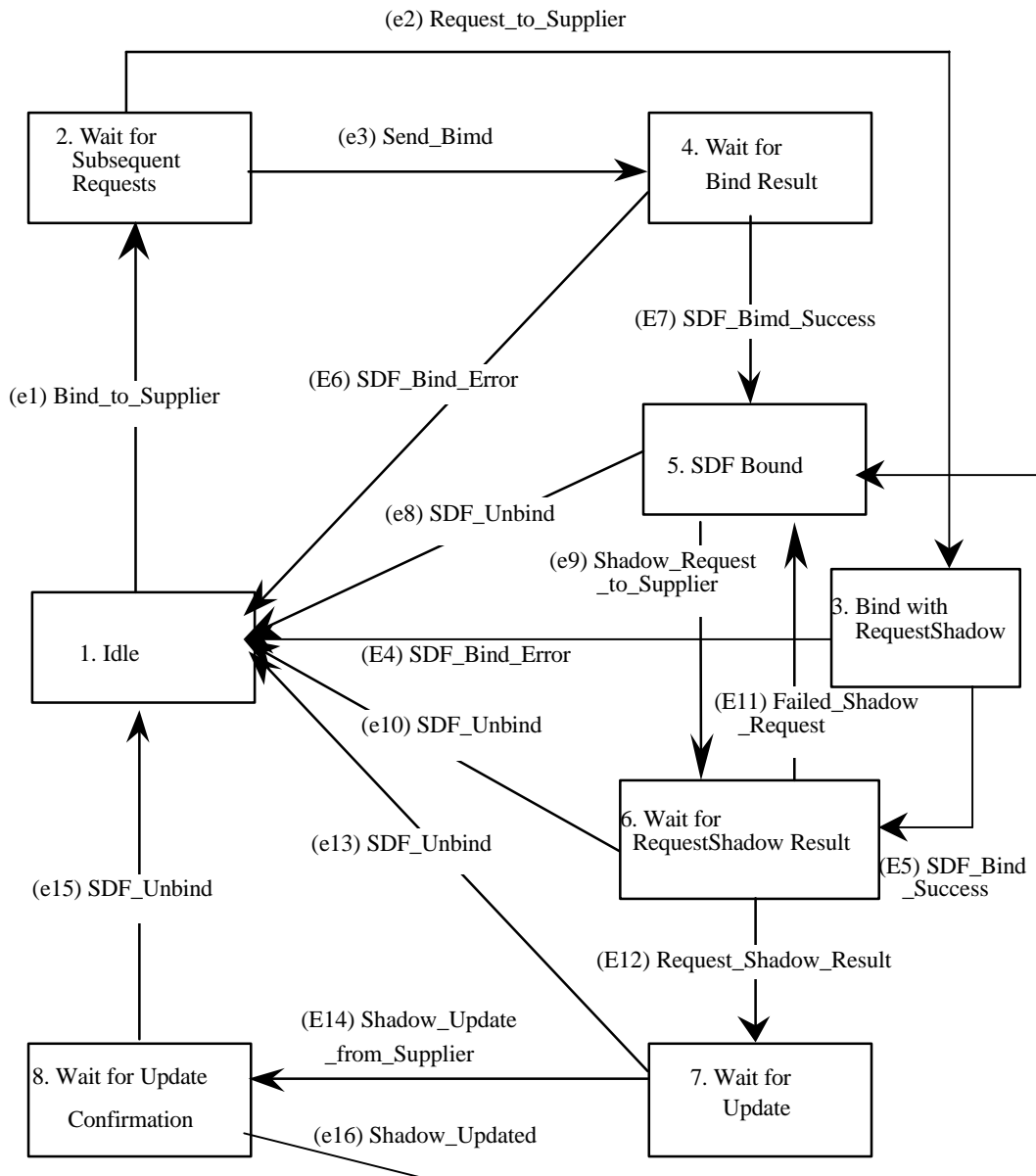


図 14 - 7 / J T - Q 1 2 2 8 シャドウ消費側起動による消費側状態機構 (SDSM-ShCCi)
(I T U - T Q . 1 2 2 8)

状態 1 : Idle

この状態で許容される唯一のイベントは以下である。

- ・ (e1) Bind_to_Supplier : これは内部イベントであり、DSAShadowBind オペレーションを送信することにより生じる。これにより状態 2、Wait for Subsequent Requests に遷移する。

状態 2 : Wait for Subsequent Requests

この状態では、DSAShadowBind オペレーションと同一メッセージで供給側に送信される RequestShadowUpdate オペレーションを待っている。以下の 2 つのイベントが考えられる。

- ・ (e2) Request_to_Supplier : これは内部イベントであり、RequestShadowUpdate オペレーションを受信す

ることにより生じる。これにより DSAShadowBind と RequestShadowUpdate オペレーションを含む TCメッセージが供給側 SDFへ送られる。これにより状態 3、Bind with RequestShadow に遷移する。

- ・ (e3) Send_Bind : これは内部イベントであり、送信すべき最終のオペレーションを受信あるいはタイマが満了したことを示すデリミタを受信することにより生じる。この内部イベントを受信されると、DSAShadowBind オペレーションを含む TCメッセージが供給側 SDF に対して送信される。これにより状態 4、Wait for Bind Result に遷移する。

状態 3 : Bind with RequestShadow

この状態では、供給側 SDF からの DSAShadowBind の結果を待っている。以下の 2 つのイベントが考えられる。

- ・ (E4) SDF_Bind_Error : これは外部イベントであり、供給側 SDF へ発行された DSAShadowBind オペレーションが失敗したことにより生じる。DSAShadowBind エラーが返送される。これにより状態 1、Idle に遷移する。
- ・ (E5) SDF_Bind_Success : これは外部イベントであり、DSAShadowBind の結果を受信することにより生じる。これは供給側 SDF へ発行した DSAShadowBind オペレーションの成功を示している。これにより状態 6、Wait for RequestShadow Result に遷移する。

状態 4 : Wait for Bind Result

この状態では、供給側 SDF からの DSAShadowBind の結果を待っている。以下の 2 つのイベントが考えられる。

- ・ (E6) SDF_Bind_Error : これは外部イベントであり、供給側 SDF へ発行された DSAShadowBind オペレーションが失敗したことにより生じる。DSAShadowBind エラーが返送される。これにより状態 1、Idle に遷移する。
- ・ (E7) SDF_Bind_Success : これは外部イベントであり、供給側 SDF へ発行された DSAShadowBind オペレーションが成功したことにより生じる。これにより状態 5、SDF Bound に遷移する。

状態 5 : SDF Bound

この状態では、消費側 SDF は供給側 SDF に対して RequestShadowUpdate オペレーションを送信する準備をしている。以下の 2 つのイベントが考えられる。

- ・ (e8) SDF_Unbind : これは内部イベントであり、2 つの SDF 間に確立した認証されたアソシエーションを終了する必要性 (例、ユーザの解放手順) あるいは、in-DSAShadowUnbind オペレーションの発行により生じる。これにより状態 1、Idle に遷移する。
- ・ (e9) Shadow_Request_to_Supplier : これは内部イベントであり、供給側 SDF に対して RequestShadowUpdate オペレーションを送信することにより生じる。これにより状態 6、Wait for RequestShadow Result に遷移する。

状態 6 : Wait for RequestShadow Result

この状態では、消費側 SDF は RequestShadowUpdate オペレーションを送信し、供給側 SDF からの応答を待っている。以下の 3 つのイベントが考えられる。

- ・ (e10) SDF_Unbind : これは内部イベントであり、2 つの SDF 間に確立した認証されたアソシエーションを終了する必要性により生じる (例、ユーザの解放手順)。これにより状態 1、Idle に遷移する。
- ・ (E11) Failed_Shadow_Request : これは外部イベントであり、発行した RequestShadowUpdate オペレ

ションに対するエラーを受信することにより生じる。これにより状態 5、SDF Bound に遷移する。

- ・ (E12) Request_Shadow_Result : これは外部イベントであり、発行した RequestShadowUpdate オペレーションに対する応答を受信することにより生じる。これにより状態 7、Wait for Update に遷移する。

状態 7 : Wait for Update

この状態では、消費側 S D F は RequestShadowUpdate の結果を受信し、供給側 S D F からの UpdateShadow オペレーションを待っている。以下の 2 つのイベントが考えられる。

- ・ (e13) SDF_Unbind : これは内部イベントであり、2 つの S D F 間に確立した認証されたアソシエーションを終了する必要性により生じる (例、ユーザの解放手順)。これにより状態 1、Idle に遷移する。
- ・ (E14) Shadow_Update_from_Supplier : これは外部イベントであり、供給側 S D F から発行された UpdateShadow オペレーションを受信することにより生じる。これにより状態 8、Wait for Update Confirmation に遷移する。

状態 8 : Wait for Update Confirmation

この状態では、消費側 S D F は供給側 S D F からの UpdateShadow オペレーションを受信し、コピーの更新を処理している。以下の 2 つのイベントが考えられる。

- ・ (e15) SDF_Unbind : これは内部イベントであり、2 つの S D F 間に確立した認証されたアソシエーションを終了する必要性により生じる (例、ユーザの解放手順)。これにより状態 1、Idle に遷移する。
- ・ (e16) Shadow_Updated : これは内部イベントであり、UpdateShadow オペレーションを正常に完了し、応答を送信することにより生じる。これにより状態 5、SDF Bound に遷移する。

14.4.2.2 連鎖に対する S D F 状態遷移モデル

連鎖手順に関しては、S D F は連鎖の起動側にも終端側にもなり得る。したがって、以下に示すように 2 つの F S M が存在する。

以下の F S M では、DSABind オペレーションと他の D S P オペレーションを一つの T C メッセージと一緒に送信する可能性を考慮している。

【連鎖起動に関するSDF状態遷移モデル(SDSM-ChI)】

SDFが連鎖の起動側になった場合に、他SDFとの相互作用を表すための有限状態機構を図14-8/JT-Q1228に示す。

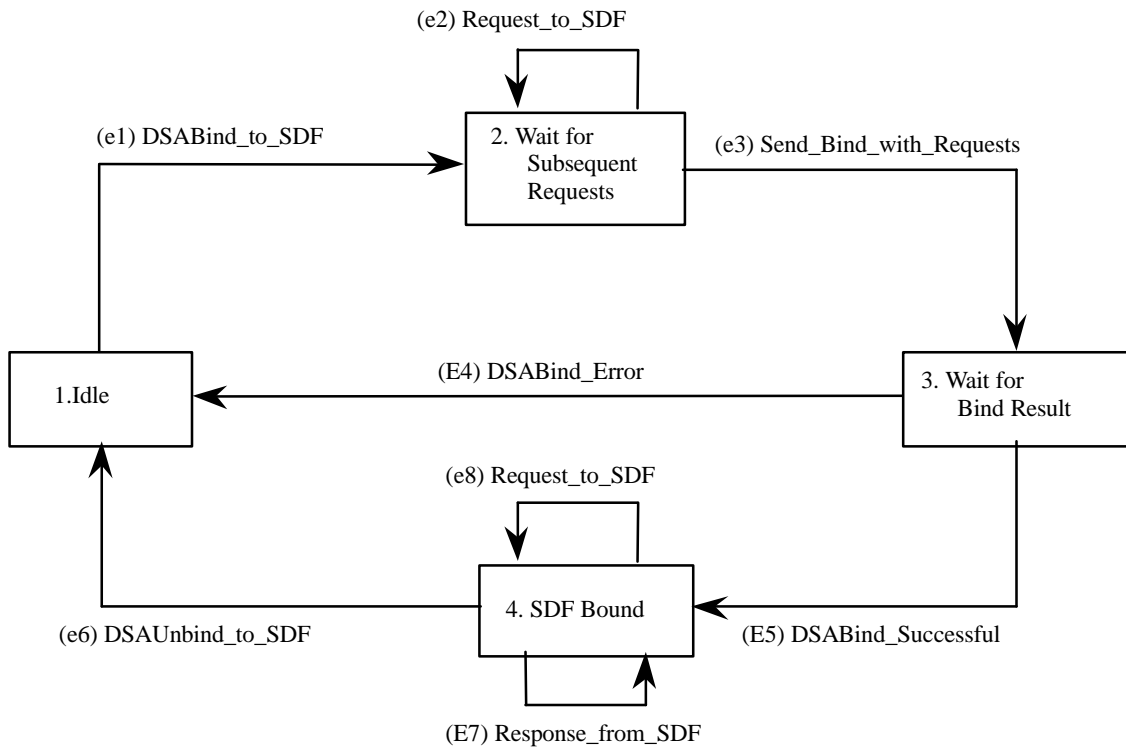


図14-8/JT-Q1228 SDF-SDF連鎖起動側の有限状態機構(SDSM-ChI)
(ITU-T Q.1228)

状態1: Idle

この状態で許容される唯一のイベントは以下である。

- ・(e1) DSABind_to_SDF: これは内部イベントであり、SDSM-ChT に対して DSABind オペレーションを送信することにより生じる。これにより状態2、Wait for Subsequent Requests に遷移する。

状態2: Wait for Subsequent Requests

この状態では、DSABind オペレーションと同一メッセージでSDSM-ChT に送信するオペレーションを待っている。以下の2つのイベントが考えられる。

- ・(e2) Request_to_SDF: これは内部イベントであり、オペレーションを受信することにより生じる。それは次のオペレーションの一つを伴う。
 - ・ chainedSearch
 - ・ chainedAddEntry
 - ・ chainedRemoveEntry
 - ・ chainedModifyEntry
 - ・ chainedExecute

オペレーションはデリミタ(あるいはタイマ満了)を受信するまでバッファリングされる。このイベントが同じ状態に遷移を起こす。

- ・ (e3) Send_Bind_with_Requests : これは内部イベントであり、送信すべき最終のオペレーションを受信したことを示すデリミタを受信することにより生じる。デリミタを受信されると、DSABind オペレーションと(もしあれば)他のオペレーションを含むメッセージが SDSM-ChT に対して送信される。これにより状態 3、Wait for Bind Results に遷移する。

状態 3 : Wait for Bind Results

この状態では、DSABind 要求が SDSM-ChT に対して送信されており、SDSM-ChT では DSABind オペレーションに関する S D F アクセス制御手順(例、アクセス認証)を実行している。以下の 2 つのイベントが考えられる。

- ・ (E4) DSABind_Error : これは外部イベントであり、SDSM-ChT に送信した DSABind オペレーションが失敗したことにより生じる。これにより状態 1、Idle に遷移する。
- ・ (E5) DSABind_Successful : これは外部イベントであり、SDSM-ChT に送信した DSABind オペレーションに対する確認を受信することにより生じる。これにより状態 4、SDF Bound に遷移する。

状態 4 : SDF Bound

この状態では、SDSM-ChI から SDSM-ChT へのアクセスが認証され、SDSM-ChT への連鎖オペレーションが送信可能となり、連鎖オペレーションに対する SDSM-ChT からの結果を受信される。以下の 3 つのイベントが考えられる。

- ・ (e6) DSAUnbind_to_SDF : これは内部イベントであり、SDSM-ChT に対して DSAUnbind オペレーションを送信することにより生じる。S D F - S D F 間のアソシエーションが終了し、関連するすべてのリソースが解放される。これにより状態 1、Idle に遷移する。
- ・ (E7) Response_from_SDF : これは外部イベントであり、SDSM-ChI により送信されたオペレーションに対する結果を受信する、あるいは SDSM-ChT からのリフェラルを受信することにより生じる。SDSM-ChI は同一の状態に留まる。
- ・ (e8) Request_to_SDF : これは内部イベントであり、SDSM-ChT に対して連鎖オペレーションを送信することにより生じる。

それは次のオペレーションの一つを伴う。

- ・ chainedSearch;
- ・ chainedAddEntry;
- ・ chainedRemoveEntry;
- ・ chainedModifyEntry
- ・ chainedExecute.

SDSM-ChI は同一の状態に留まる。

【連鎖終端に関する S D F 状態遷移モデル (SDSM-ChT)】

S D F が連鎖の終端側になった場合に、他 S D F との相互作用を表すための有限状態機構を図 1 4 - 9 / J T - Q 1 2 2 8 に示す。

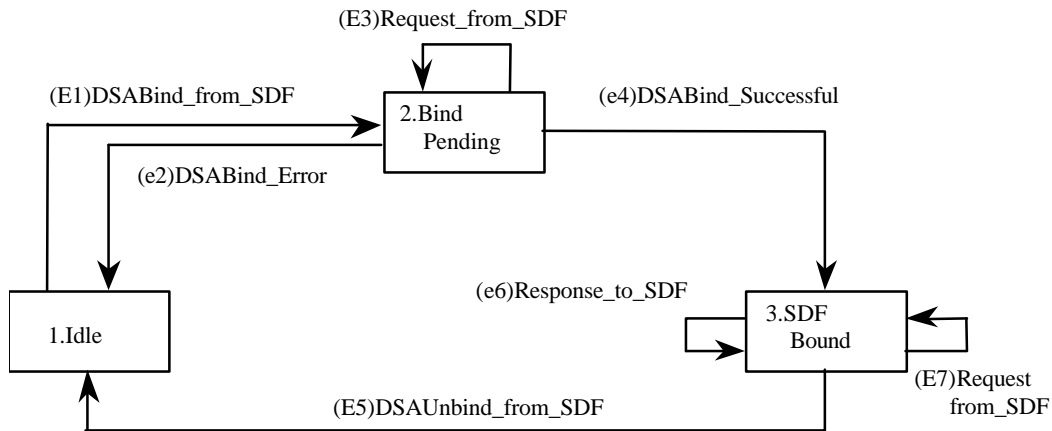


図 1 4 - 9 / J T - Q 1 2 2 8 S D F - S D F 連鎖終端側の有限状態機構 (SDSM-ChT)
(I T U - T Q . 1 2 2 8)

状態 1 : Idle

この状態で許容される唯一のイベントは以下である。

- ・ (E 1) DSABind_from_SDF : これは外部イベントであり、SDSM-ChI からの DSABind オペレーションを受信することにより生じる。これにより状態 2、Bind Pending に遷移する。

状態 2 : Bind Pending

この状態では、DSABind 要求を SDSM-ChI から受信しており、SDSM-ChT では DSABind オペレーションに関する S D F アクセス制御手順 (例、アクセス認証) を実行している。以下の 2 つのイベントが考えられる。

- ・ (e 2) DSABind_Error : これは内部イベントであり、SDSM-ChI から送信された DSABind オペレーションが失敗したことにより生じる。これにより状態 1、Idle に遷移し、結合エラーが SDSM-ChI に返送される。
- ・ (E 3) Request_from_SDF : これは外部イベントであり、DSABind オペレーションの結果が判明する前にオペレーションを受信することにより生じる。それは次のオペレーションの一つを伴う。
 - ・ chainedSearch;
 - ・ chainedAddEntry;
 - ・ chainedRemoveEntry;
 - ・ chainedModifyEntry
 - ・ chainedExecute.

オペレーションは記憶され、そして SDSM-ChT は同一状態に留まる。別の状態に遷移が起これるとき、そのオペレーションはそれらがその状態に起こったかのように再び見られる。

- ・ (e 4) DSABind_Successful : これは内部イベントであり、SDSM-ChT に送信された DSABind オペレーションが成功することにより生じる。これにより状態 3、SDF Bound に遷移する。

状態 3 : SDF Bound

この状態では、SDSM-ChI から SDSM-ChT へのアクセスが認証され、SDSM-ChI から送信される連鎖オペレーションが許容される。SDSM-ChI からの要求を待つことに加え、SDSM-ChT は受信したオペレーションに対する応答を本状態で送信できる。以下の 3 つのイベントが考えられる。

- ・ (E5) DSAUnbind_from_SDF : これは外部イベントであり、SDSM-ChI からの DSAUnbind オペレーションを受信することにより生じる。SDF - SDF 間のアソシエーションが終了し、関連するすべてのリソースが解放される。これにより状態 1、Idle に遷移する。
- ・ (e6) Response_to_SDF : これは内部イベントであり、SDSM-ChI から受信したオペレーションが完了する、あるいは SDSM-ChI へのリフェラルを生成することにより生じる。応答あるいはリフェラルが SDSM-ChI に送信される。SDSM-ChT は同一の状態に留まる。
- ・ (E7) Request_from_SDF : これは外部イベントであり、SDSM-ChI からの要求を受信することにより生じる。

それは次のオペレーションの一つを伴う。

- ・ chainedSearch
- ・ chainedAddEntry
- ・ chainedRemoveEntry
- ・ chainedModifyEntry
- ・ chainedExecute

SDSM-ChT は同一の状態に留まる。

15 . C U S F アプリケーションエンティティ手順

#

16 . エラー手順

本節は、INCS - 1INAPに対する一般的なエラー手順を定義する。エラー手順は二つの節に分けて記述される。16.1 節はINAPオペレーションに関連したエラーを、16.2 節はINAPオペレーションと直接関連しない異なったFEのエラー条件に関連したエラーを記述している。

16.1 オペレーション関連エラー手順

以下の節では、オペレーションに関連するエラーに対する一般的なエラー手順を定義する。エラーはオペレーションエラーとして4章、7章、8章に定義される。オペレーションエラーを報告するのに使用されるTCサービスは18.1 節で記述される。

個々のオペレーションに対する固有の手順を有するエラーは11章から15章に関連するオペレーションの詳細手順とともに示される。

16.1.1 AttributeError

16.1.1.1 一般記述

16.1.1.1.1 エラー記述

このエラーは、属性関連の問題を報告するためにSDFからSCFあるいはSDFに送信される。AttributeErrorが送信される条件は、ITU - T勧告X . 5 1 1の12.4 節に定義されている。

16.1.1.1.2 アーギュメント記述

AttributeErrorのパラメータ及び問題コードは、ITU - T勧告X . 5 1 1の12.4 節に定義されている。

16.1.1.2 SCF->SDFオペレーション

AddEntry

Execute

ModifyEntry

Search

起動側エンティティ(SCF)での手順

A) オペレーション送信

前条件： SCSM 状態 4 SDF Bound あるいは、
SCSM 状態 2 Wait for subsequent requests

後条件： SCSM 状態 4 SDF Bound あるいは、
SCSM 状態 2 Wait for subsequent requests

B) エラー受信

前条件： SCSM 状態 4 SDF Bound

後条件： SCSM 状態 4 SDF Bound

エラー手順はサービス論理に依存する。SCFが要求を変更することが可能であれば、別のSDFへの問い合わせを行うことが可能である。そうでなければサービス処理は終了すべきである。

応答側エンティティ(SDF)での手順

A) オペレーション受信

前条件： SDF FSM 状態 3 SCF Bound

後条件： SDF FSM 状態 3 SCF Bound

B) エラー返送

前条件： SDF FSM 状態 3 SCF Bound

後条件： SDF FSM 状態 3 SCF Bound

SDF は属性の問題のためオペレーションを実行できない。それゆえ SCF に AttributeError を返送する。エラーを返送した後、行われるエラー手順はない。

16.1.1.3 SDF-->SDF オペレーション

ChainedAddEntry

ChainedExecute

ChainedModifyEntry

ChainedSearch

起動側エンティティ(SDF)での手順

A) オペレーション送信

前条件： SDSM-ChI 状態 4 SDF Bound あるいは、
SDSM-ChI 状態 2 Wait for subsequent requests

後条件： SDSM-ChI 状態 4 SDF Bound あるいは、
SDSM-ChI 状態 2 Wait for subsequent requests

B) エラー受信

前条件： SDSM-ChI 状態 4 SDF Bound

後条件： SDSM-ChI 状態 4 SDF Bound

本エラーは SCF へ通知される。

応答側エンティティ(SDF)での手順

A) オペレーション受信

前条件： SDSM-ChT 状態 3 SDF Bound

後条件： SDSM-ChT 状態 3 SDF Bound

B) エラー返送

前条件： SDSM-ChT 状態 3 SDF Bound

後条件： SDSM-ChT 状態 3 SDF Bound

SDF は属性の問題のためオペレーションを実行できない。それゆえ他の SDF に AttributeError を返送する。エラーを返送した後、行われるエラー手順はない。

16.1.2 Canceled

#

16.1.3 CancelFailed

#

16.1.4 DSAReferral

16.1.4.1 一般記述

16.1.4.1.1 エラー記述

このエラーは、SDF によって別の SDF へ一連の連鎖オペレーションに関連した問題を通知する。DSAReferral が返送される条件は、ITU - T 勧告 X . 5 1 8 の 8.3 節に定義されている。

16.1.4.1.2 アーギュメント記述

DSAReferral のパラメータ及び問題コードは、ITU - T 勧告 X . 5 1 8 の 13.2 節に定義されている。

16.1.4.2 S D F --> S D F オペレーション

CainedAddEntry
ChainedExecute
ChainedModifyEntry
ChainedRemoveEntry
ChainedSearch

起動側エンティティ(S D F)での手順

A) オペレーション送信

前条件： SDSM-ChI 状態 4 SDF Bound あるいは、
SDSM-ChI 状態 2 Wait for subsequent requests
後条件： SDSM-ChI 状態 4 SDF Bound あるいは、
SDSM-ChI 状態 2 Wait for subsequent requests

B) エラー受信

前条件： SDSM-ChI 状態 4 SDF Bound
後条件： SDSM-ChI 状態 4 SDF Bound

DSAReferral エラー受信後、S D Fはこのエラーにより指示された別のS D Fにアクセスすることによりオペレーションの実行を継続できる。

応答側エンティティ(S D F)での手順

A) オペレーション受信

前条件： SDSM-ChT 状態 3 SDF Bound
後条件： SDSM-ChT 状態 3 SDF Bound

B) エラー返送

前条件： SDSM-ChT 状態 3 SDF Bound
後条件： SDSM-ChT 状態 3 SDF Bound

S D Fは chaining 条件のためオペレーションを実行できない。それゆえ別のS D Fに DSAReferral エラーを返送する。エラーを返送した後、行われるエラー手順はない。

16.1.5 ETCFailed

#

16.1.6 Execution エラー

16.1.6.1 一般記述

16.1.6.1.1 エラー記述

Execution エラーはS D FからS C Fへ送られるエラーであり、エントリメソッドを実行するための要求が失敗したことを示す。この失敗は不正入力値、内部オペレーションもしくはエントリメソッドの実行に関連したデータアクセス論理の失敗に対するものである。

16.1.6.1.2 アーギュメント記述

```
PARAMETER OPTIONALLY-PROTECTED{
    SET{
        problem[0] ExecutionProblem,
        COMPONENTS OF CommonResults},
    DIRQOP.&dirErrors-QOP{ @dirqop}}
ExecutionProblem ::= INTEGER{
    missingInputValue(1),
    executionFailure(2)}
```

16.1.6.2 S C F --> S D F オペレーション

Execute

起動側エンティティ (S C F) での手順

A) オペレーション送信

前条件： SCSM 状態 2 Wait for subsequent requests あるいは、
SCSM 状態 4 SDF Bound

後条件： SCSM 状態 2 Wait for subsequent requests あるいは、
SCSM 状態 4 SDF Bound

B) エラー受信

前条件： SCSM 状態 4 SDF Bound

後条件： SCSM 状態 4 SDF Bound

エラー手順はサービス論理に非依存である。サービス処理は終了すべきである。

応答側エンティティ (S D F) での手順

A) オペレーション受信

前条件： SDF FSM 状態 3 SCF Bound

後条件： SDF FSM 状態 2 Bind Pending あるいは、
SDF FSM 状態 3 SCF Bound

B) エラー返送

前条件： SDF FSM 状態 3 SCF Bound

後条件： SDF FSM 状態 3 SCF Bound

S D F が正しい入力値を受信できなかった場合には、S D F は missingInputValues を設定された problem により S C F に対して ExecusionError を送信する。その他全ての場合は、内部オペレーションもしくは、データアクセス論理の不具合による失敗がなければ、S D F はエントリメソッドの実行を試みる。それゆえ S D F は executionFailure を設定された problem により S C F に対して ExecutionError を送信する。エラー返送後、S D F は、そのオペレーションの前の状態に戻される。すなわち、その execute オペレーションは一つの原子オペレーションと考えられる。

16.1.6.3 S D F --> S D F オペレーション

ChainedExecute

起動側エンティティ (S D F) での手順

A) オペレーション送信

前条件： SDSM-ChI 状態 4 SDF Bound あるいは、
状態 2 Wait for subsequent requests

後条件： SDSM-ChI 状態 4 SDF Bound あるいは、
状態 2 Wait for subsequent requests

B) エラー受信

前条件： SDSM-ChI 状態 4 SDF Bound

後条件： SDSM-ChI 状態 4 SDF Bound

このエラーは、Chained オペレーションの場合に S C F へ報告される。

応答側エンティティ (S D F) での手順

A) オペレーション受信

前条件： SDSM-ChT 状態 3 SDF Bound

後条件： SDSM-ChT 状態 3 SDF Bound

B) エラー返送

前条件： SDSM-ChT 状態 3 SDF Bound

後条件： SDSM-ChT 状態 3 SDF Bound

S D F がそのオペレーションを実行できなければ、他の S D F に ExecutionError を送信する。エラー返送後、それ以上のエラー処理は実行されない。

16.1.7 ImproperCallerResponse #

16.1.8 MissingCustomerRecord #

16.1.9 MissingParameter #

16.1.10 NameError

16.1.10.1 一般記述

16.1.10.1.1 エラー記述

このエラーは、オブジェクトの名前に関する問題を報告するため、S D F から S C F もしくは別の S D F へ送られる。

NameError が出される状態は、I T U - T 勧告 X . 5 1 1 12.5 節で定義されている。

16.1.10.1.2 アーギュメント記述

この name エラーパラメータと problem パラメータは I T U - T 勧告 X . 5 1 1 12.5 節に明記される。

16.1.10.2 S C F --> S D F オペレーション

AddEntry

Execute

ModifyEntry

RemoveEntry

Search

起動側エンティティ (S C F) における手順

A) オペレーション送信

前条件： SCSM 状態 4 SDF Bound あるいは、
SCSM 状態 2 Wait for subsequent requests

後条件： SCSM 状態 4 SDF Bound あるいは、
SCSM 状態 2 Wait for subsequent requests

B) エラー受信

前条件： SCSM 状態 4 SDF Bound

後条件： SCSM 状態 4 SDF Bound

エラー手順は、サービス論理に依存する。S C F が要求を変更することが可能ならば、別の S D F に問い合わせを行うことができる。そうでない場合、サービス処理を終えるべきである。

応答側エンティティ (S D F) における手順

A) オペレーション受信

前条件： SDF FSM 状態 3 SCF Bound

後条件： SDF FSM 状態 3 SCF Bound

B) エラー返送

前条件： SDF FSM 状態 3 SCF Bound

後条件： SDF FSM 状態 3 SCF Bound

S D F は、name 問題のためにオペレーションを実行出来ない。従って S C F へ Name エラーを送信する。エラー返送後、更なるエラー処理は、実行されない。

16.1.10.3 S D F --> S D F オペレーション

ChainedAddEntry

ChainedExecute

ChainedModifyEntry

ChainedRemoveEntry

ChainedSearch

起動側エンティティ (S D F) における手順

A) オペレーション送信

前条件： SDSM-ChI 状態 4 SDF Bound あるいは、
SDSM-ChI 状態 2 Wait for subsequent requests

後条件： SDSM-ChI 状態 4 SDF Bound あるいは、
SDSM-ChI 状態 2 Wait for subsequent requests

B) エラー受信

前条件： SDSM-ChI 状態 4 SDF Bound

後条件： SDSM-ChI 状態 4 SDF Bound

このエラーは、S C F に通知される。

応答側エンティティ (S D F) における手順

A) オペレーション受信

前条件： SDSM-ChT 状態 3 SDF Bound

後条件： SDSM-ChT 状態 3 SDF Bound

B) エラー受信

前条件： SDSM-ChT 状態 3 SDF Bound

後条件： SDSM-ChT 状態 3 SDF Bound

SDF は、name 問題のためにオペレーションを実行出来ない。従って他の SDF にある Name エラーを送信する。エラー返信後、更なるエラー処理は、実行されない。

16.1.11 ParameterOutOfRange

#

16.1.12 Referral

16.1.12.1 一般記述

16.1.12.1.1 エラー記述

このエラーは、SDF から SCF へ送信され、サービスデータロケーションに関する problem を報告する。Referral エラーが出される条件は、ITU - T 勧告 X . 5 1 1 12.6 節に定義される。

16.1.12.1.2 アーギュメント記述

referral エラーパラメータと problem コードは、ITU - T 勧告 X . 5 1 1 12.6 節に明記される。

16.1.12.2 SCF --> SDF オペレーション

AddEntry

Execuse

ModifyEntry

RemoveEntry

Search

起動側エンティティ (SCF) における手順

A) オペレーション送信

前条件： SCSM 状態 4 SDF Bound あるいは、
SCSM 状態 2 Wait for subsequent requests

後条件： SCSM 状態 4 SDF Bound あるいは、
SCSM 状態 2 Wait for subsequent requests

B) エラー受信

前条件： SCSM 状態 4 SDF Bound

後条件： SCSM 状態 4 SDF Bound

Referral エラー受信後、SCF は、このエラーで指示された別の SDF をアクセスすることにより、サービス論理の実行を継続できる。

応答側エンティティ (SDF) における手順

A) オペレーション受信

前条件： SDF FSM 状態 3 SCF Bound

後条件： SDF FSM 状態3 SCF Bound

B) エラー返送

前条件： SDF FSM 状態3 SCF Bound

後条件： SDF FSM 状態3 SCF Bound

SDFは、データロケーションに従うオペレーションを実行できない。従ってSCFへReferralエラーを送信する。エラー返送後、更なるエラー処理は実行されない。

16.1.13 RequestedInfoError #

16.1.14 ScfReferral #

16.1.15 Security

16.1.15.1 一般記述

16.1.15.1.1 エラー記述

このエラーは、セキュリティの理由でオペレーションを実行した場合の問題を報告するためにSDFから、SCFあるいはSDFに送信される。securityErrorが送信される条件は、ITU-T勧告X.511の12.7節に定義されている。

16.1.15.1.2 アーギュメント記述

securityErrorのパラメータ及び問題コードは、ITU-T勧告X.511の12.7節に定義されている。

16.1.15.2 SCF-->SDFオペレーション

AddEntry

Execute

ModifyEntry

RemoveEntry

Search

起動側エンティティ(SCF)での手順

A) オペレーション送信

前条件： SCSM 状態2 Wait for subsequent requests あるいは、
SCSM 状態4 SDF Bound

後条件： SCSM 状態2 Wait for subsequent requests あるいは、
SCSM 状態4 SDF Bound

B) エラー受信

前条件： SCSM 状態3 Wait for Bind result あるいは、
SCSM 状態4 SDF Bound

後条件： SCSM 状態4 SDF Bound

エラー手順はサービス論理に非依存である。サービス処理は終了すべきである。

応答側エンティティ(SDF)での手順

A) オペレーション受信

前条件： SDF FSM 状態1 Idle あるいは、

	SDF FSM 状態 3	SCF Bound
後条件 :	SDF FSM 状態 2	Bind Pending あるいは、
	SDF FSM 状態 3	SCF Bound

B) エラー返送

前条件 :	SDF FSM 状態 2	Bind Pending あるいは、
	SDF FSM 状態 3	SCF Bound
後条件 :	SDF FSM 状態 3	SCF Bound

セキュリティ上の理由から S D F はオペレーションを実行できないため、S C F に securityError を返送する。
エラーを返送した後、行われるエラー手順はない。

16.1.15.3 S D F --> S D F オペレーション

ChainedAddEntry
ChainedExecute
ChainedModifyEntry
ChainedRemoveEntry
ChainedSearch

起動側エンティティ(S D F)での手順

A) オペレーション送信

前条件 :	SDSM-ChI 状態 4	SDF Bound あるいは、
	SDSM-ChI 状態 2	Wait for subsequent requests
後条件 :	SDSM-ChI 状態 4	SDF Bound あるいは、
	SDSM-ChI 状態 2	Wait for subsequent requests

B) エラー受信

前条件 :	SDSM-ChI 状態 4	SDF Bound
後条件 :	SDSM-ChI 状態 4	SDF Bound

chained オペレーションの場合、本エラーは S C F へ通知される。

応答側エンティティ(S D F)での手順

A) オペレーション受信

前条件 :	SDSM-ChT 状態 3	SDF Bound
後条件 :	SDSM-ChT 状態 3	SDF Bound

B) エラー返送

前条件 :	SDSM-ChT 状態 3	SDF Bound
後条件 :	SDSM-ChT 状態 3	SDF Bound

セキュリティ上の理由から S D F はオペレーションを実行できないため、S D F に securityError を返送する。
エラーを返送した後、行われるエラー手順はない。

16.1.15.4 S C F --> S C F オペレーション

#

16.1.16 Service

16.1.16.1 一般記述

16.1.16.1.1 エラー記述

このエラーは、サービスの提供に関連した問題を報告するために S D F から、S C F あるいは S D F に送信される。serviceError が送信される条件は、I T U - T 勧告 X . 5 1 1 の 12.8 節に定義されている。

16.1.16.1.2 アーギュメント記述

serviceError のパラメータ及び問題コードは、I T U - T 勧告 X . 5 1 1 の 12.8 節に定義されている。

16.1.16.2 S C F --> S D F オペレーション

AddEntry

Execute

ModifyEntry

RemoveEntry

Search

起動側エンティティ(S C F)での手順

A) オペレーション送信

前条件： SCSM 状態 2 Wait for subsequent requests あるいは、
SCSM 状態 4 SDF Bound

後条件： SCSM 状態 2 Wait for subsequent requests あるいは、
SCSM 状態 4 SDF Bound

B) エラー受信

前条件： SCSM 状態 4 SDF Bound

後条件： SCSM 状態 4 SDF Bound

エラー手順はサービス論理に依存する。もし、代替の S D F があれば、別の問い合わせを行うことが可能である。そうでなければサービス処理は終了すべきである。

応答側エンティティ(S D F)での手順

A) オペレーション受信

前条件： SDF FSM 状態 3 SDF Bound

後条件： SDF FSM 状態 3 SDF Bound

B) エラー返送

前条件： SDF FSM 状態 2 Bind Pending (Bind 時)

SDF FSM 状態 3 SCF Bound (Bind 以外のオペレーション時)

後条件： SDF FSM 状態 1 Idle (Bind 時)

SDF FSM 状態 3 SCF Bound (Bind 以外のオペレーション時)

S D F はサービスに関連する問題によりオペレーションを実行できないため、S C F に serviceError を返送する。エラーを返送した後、行われるエラー手順はない。

16.1.16.3 S D F --> S D F オペレーション

chainedAddEntry

chainedExecute
chainedModifyEntry
chainedRemoveEntry
chainedSearch

起動側エンティティ(SDF)での手順

A) オペレーション送信

前条件: SDSM-ChI 状態 4 SDF Bound あるいは、
SDSM-ChI 状態 2 Wait for subsequent requests

後条件: SDSM-ChI 状態 4 SDF Bound あるいは、
SDSM-ChI 状態 2 Wait for subsequent requests

B) エラー受信

前条件: SDSM-ChI 状態 4 SDF Bound

後条件: SDSM-ChI 状態 4 SDF Bound

chained オペレーションの場合、本エラーはSCFへ通知される。

応答側エンティティ(SDF)での手順

A) オペレーション受信

前条件: SDSM-ChT 状態 3 SDF Bound

後条件: SDSM-ChT 状態 3 SDF Bound

B) エラー返送

前条件: SDSM-ChT 状態 3 SDF Bound

後条件: SDSM-ChT 状態 3 SDF Bound

SDFはサービスに関連する問題によりオペレーションを実行できないため、SDFに serviceError を返送する。エラーを返送した後、行われるエラー手順はない。

16.1.17 Shadow

16.1.17.1 一般記述

16.1.17.1.1 エラー記述

このエラーはシャドウイングに関する問題の報告のため、SDFからSDFへ送信される。shadowError が返送される条件は、ITU-T勧告X.525の12章に定義されている。

16.1.17.1.2 アーギュメント記述

shadowError のパラメータ及び問題コードは、ITU-T勧告X.525の11.3.3節に定義されている。

16.1.17.2 SDF-->SDFオペレーション

CoordinateShadowUpdate

RequestShadowUpdate

UpdateShadow

供給側エンティティ(SDF)での手順

A-1) オペレーション送信

前条件： SDSM-ShSSi (供給側起動の場合)

- 状態 8 SDF Bound (CoordinateShadowUpdate の場合) あるいは、
- 状態 2 Wait for Subsequent Requests (CoordinateShadowUpdate の場合) あるいは、
- 状態 10 Wait for Update (UpdateShadow の場合) あるいは、
- 状態 3 Bind with Coordinate Shadow (UpdateShadow の場合)

SDSM-ShSCi (消費側起動の場合)

- 状態 5 Wait for Update (UpdateShadow の場合) あるいは、
- 状態 4 Wait for RequestShadow Result (UpdateShadow の場合)

後条件： SDSM-ShSSi (供給側起動の場合)

- 状態 9 Wait for Coordination Result (CoordinateShadowUpdate の場合) あるいは、
- 状態 3 Bind with Coordinate Shadow (CoordinateShadowUpdate の場合) あるいは、
- 状態 11 Wait for Update Confirmation (UpdateShadow の場合) あるいは、
- 状態 5 Bind with CoordinateShadow and Update (UpdateShadow の場合)

SDSM-ShSCi (消費側起動の場合)

- 状態 6 Wait for Update Confirmation (UpdateShadow の場合)

B-1) エラー受信

前条件： SDSM-ShSSi (供給側起動の場合)

- 状態 9 Wait for Coordination Result (CoordinateShadowUpdate の場合) あるいは、
- 状態 6 Bound with Coordinate Shadow Sent (CoordinateShadowUpdate の場合) あるいは、
- 状態 11 Wait for Update Confirmation (UpdateShadow の場合)

SDSM-ShSCi (消費側起動の場合)

- 状態 6 Wait for Update Confirmation(Update Shadow の場合)

後条件： SDSM-ShSSi (供給側起動の場合)

- 状態 8 SDF Bound (CoordinateShadowUpdate あるいは UpdateShadow の場合)

SDSM-ShSCi (消費側起動の場合)

- 状態 3 SDF Bound (UpdateShadow の場合)

A-2) オペレーション受信

前条件： SDSM-ShSCi (消費側起動の場合)

- 状態 3 SDF Bound(RequestShadowUpdate の場合)

後条件： SDSM-ShSCi (消費側起動の場合)

- 状態 4 Wait for RequestShadow Result (RequestShadowUpdate の場合)

B-2) エラー返送

前条件： SDSM-ShSCi (消費側起動の場合)

- 状態 4 Wait for RequestShadow Result (RequestShadowUpdate の場合)

後条件： SDSM-ShSCi (消費側起動の場合)

- 状態 3 SDF Bound (RequestShadowUpdate の場合)

本エラーの受信または返送後、行われるエラー手順はない。

消費側エンティティ(SDF)での手順

A-1) オペレーション送信

前条件： SDSM-ShCCi (消費側起動の場合)

状態 5 SDF Bound (RequestShadowUpdate の場合) あるいは、
状態 2 Wait for Subsequent Requests (RequestShadowUpdate の場合)
後条件 : SDSM-ShCCi (消費側起動の場合)
状態 6 Wait for RequestShadow Result (RequestShadowUpdate の場合) あるいは、
状態 3 Bind with RequestShadow (RequestShadowUpdate の場合)

B-1) エラー受信

前条件 : SDSM-ShCCi (RequestShadowUpdate の場合)
状態 6 Wait for Requestshadow Result (RequestShadowUpdate の場合)
後条件 : SDSM-ShCCi (RequestShadowUpdate の場合)
状態 5 SDF Bound (RequestShadowUpdate の場合)

A-2) オペレーション受信

前条件 : SDSM-ShCSi (供給側起動の場合)
状態 3 SDF Bound (CoordinateShadowUpdate の場合) あるいは、
状態 5 Wait for Update (UpdateShadow の場合)
SDSM-ShCCi (消費側起動の場合)
状態 7 Wait for Update(UpdateShadow の場合)
後条件 : SDSM-ShCSi (供給側起動の場合)
状態 4 Wait for Coordination result (CoordinateShadowUpdate の場合) あるいは、
状態 6 Wait for Update Confirmation (UpdateShadow の場合)
SDSM-ShCCi (消費側起動の場合)
状態 8 Wait for Update Confirmation (UpdateShadow の場合)

B-2) エラー返送

前条件 : SDSM-ShCSi (供給側起動の場合)
状態 4 Wait for Coordination Result (CoordinateShadowUpdate の場合) あるいは、
状態 6 Wait for Update Confirmation (UpdateShadow の場合)
SDSM-ShCCi (消費側起動の場合)
状態 8 Wait for Update Confirmation (UpdateShadow の場合)
後条件 : SDSM-ShCSi (供給側起動の場合)
状態 3 SDF Bound (CoordinateShadowUpdate あるいは UpdateShadow の場合)
SDSM-ShCCi (消費側起動の場合)
状態 5 SDF Bound (UpdateShadow の場合)

本エラーの受信または返送後、行われるエラー手順はない。

16.1.18	SystemFailure	#
16.1.19	TaskRefused	#
16.1.20	UnavailableResource	#
16.1.21	UnexpectedComponentSequence	#
16.1.22	UnexpectedDataValue	#
16.1.23	UnexpectedParameter	#
16.1.24	UnknownLegID	#
16.1.25	UnknownResource	#

16.1.26 Update

16.1.26.1 一般記述

16.1.26.1.1 エラー記述

このエラーは、SDF中での情報の追加、削除、あるいは更新に関連した問題を報告するためにSDFからSCFあるいはSDFに送信される。updateError が送信される条件は、ITU-T勧告X.511の12.9節に定義されている。

16.1.26.1.2 アーギュメント記述

updateErrorのパラメータ及び問題コードは、ITU-T勧告X.511の12.9節に定義されている。

16.1.26.2 SCF->SDFオペレーション

AddEntry

Execute

ModifyEntry

RemoveEntry

起動側エンティティ(SCF)での手順

A) オペレーション送信

前条件： SCSM 状態4 SDF Bound あるいは、
SCSM 状態2 Wait for subsequent requests

後条件： SCSM 状態4 SDF Bound あるいは、
SCSM 状態2 Wait for subsequent requests

B) エラー受信

前条件： SCSM 状態4 SDF Bound

後条件： SCSM 状態4 SDF Bound

エラー手順はサービス論理に依存する。

応答側エンティティ(SDF)での手順

A) オペレーション受信

前条件： SDF FSM 状態3 SCF Bound

後条件： SDF FSM 状態3 SCF Bound

B) エラー返送

前条件： SDF FSM 状態3 SCF Bound

後条件： SDF FSM 状態3 SCF Bound

SDFは情報の追加、削除あるいは更新に関連する問題によるオペレーションを実行できないため、SCFにupdateErrorを返送する。エラーを返送した後、行われるエラー手順はない。

16.1.26.3 SDF-->SDFオペレーション

chainedAddEntry
chainedExecute
chainedModifyEntry
chainedRemoveEntry

起動側エンティティ(SDF)での手順

A) オペレーション送信

前条件：	SDSM-ChI 状態4	SDF Bound あるいは、
	SDSM-ChI 状態2	Wait for subsequent requests
後条件：	SDSM-ChI 状態4	SDF Bound あるいは、
	SDSM-ChI 状態2	Wait for subsequent requests

B) エラー受信

前条件：	SDSM-ChI 状態4	SDF Bound
後条件：	SDSM-ChI 状態4	SDF Bound

本エラーはSCFへ通知される。

応答側エンティティ(SDF)での手順

A) オペレーション受信

前条件：	SDSM-ChT 状態3	SDF Bound
後条件：	SDSM-ChT 状態3	SDF Bound

B) エラー返送

前条件：	SDSM-ChT 状態3	SDF Bound
後条件：	SDSM-ChT 状態3	SDF Bound

SDFは情報の追加、削除あるいは更新に関連する問題によるオペレーションを実行できないため、他のSCFにupdateErrorを返送する。エラーを返送した後、行われるエラー手順はない。

16.1.27 chainingRefused

#

16.1.28 DirectoryBindError

16.1.28.1 一般記述

16.1.28.1.1 エラー記述

このエラーは正しい関係を立証した問題を報告するためにSDFからSCF/SDFへ送られる。directoryBindErrorの条件はITU-T勧告X.511に記載されている。

16.1.28.1.2 アーギュメント記述

directoryBindErrorのパラメータおよび問題コードはITU-T勧告X.511に記載されている。

16.1.28.2 S C F --> S D F オペレーション

directoryBind

起動側エンティティ(S D F)での手順

A) オペレーション送信

前条件: SCSM 状態 1 Idle

後条件: SCSM 状態 3 Wait for Bind result

B) エラー受信

前条件: SCSM 状態 3 Wait for Bind result

後条件: SCSM 状態 1 Idle

エラー手順はサービス論理から独立している。サービスプロセスは終了されるべきである。

応答側エンティティ(S D F)での手順

A) オペレーション受信

前条件: SDF FSM 状態 1 Idle

後条件: SDF FSM 状態 2 Bind Pending

B) エラー返送

前条件: SDF FSM 状態 2 Bind Pending (Bind 時)

後条件: SDF FSM 状態 1 Idle (Bind 時)

S D F はオペレーションを実行できないため directoryBindError を S C F に送信する。エラーを返送した後、行われるエラー手順はない。

16.1.28.3 S D F --> S D F オペレーション

DSABind

DSAShadowBind

起動側エンティティ(S D F)での手順

A) オペレーション送信

前条件: SDSM-ChI 状態 1 Idle (DSABind の場合) あるいは、

SDSM-ShSSi 状態 1 Idle (供給側起動 DSAShadowBind の場合) あるいは、

SDSM-ShCCi 状態 1 Idle (消費側起動 DSA ShadowBind の場合)

後条件: SDSM-ChI 状態 2 Wait for subsequent requests (DSABind の場合) あるいは、

SDSM-ShSSi 状態 2 Wait for subsequent requests (供給側起動 DSAShadowBind の場合) あるいは、

SDSM-ShCCi 状態 2 Wait for subsequent requests (消費側起動 DSAShadowBind の場合)

B) エラー受信

前条件: SCSM-ChI 状態 3 Wait for Bind result (DSABind の場合) あるいは、

SDSM-ShSCi 状態 2 Wait for Bind result (供給側起動 DSAShadowBind の場合) あるいは、

SDSM-ShCCi 状態 4 Wait for Bind result (消費側起動 DSAShadowBind の場合)

後条件: SDSM-ChI 状態 1 Idle (DSABind の場合) あるいは、

SDSM-ShSCi 状態 1 Idle (供給側起動 DSAShadowBind の場合) あるいは、

SDSM-ShCCi 状態 1 Idle (消費側起動 DSAShadowBind の場合)

このエラーは DSABind オペレーションの場合 S C F へ送信される。

応答側エンティティ(S D F)での手順

A) オペレーション受信

前条件: SDSM-ChT 状態 1 Idle (DSABind の場合) あるいは、
SDSM-ShCSi 状態 1 Idle (供給側起動 DSAShadowBind の場合) あるいは、
SDSM-ShSCi 状態 1 Idle (消費側起動 DSAShadowBind の場合)
後条件: SDSM-ChT 状態 2 Bind Pending (DSAShadowBind の場合) あるいは、
SDSM-ShCSi 状態 2 Wait for Bind result (供給側起動 DSAShadowBind の場合) あるいは、
SDSM-ShSCi 状態 2 Wait for Bind result (消費側起動 DSAShadowBind の場合)

B) エラー返送

前条件: SCSM-ChT 状態 2 Bind Pending (DSABind の場合) あるいは、
SDSM-ShCSi 状態 2 Wait for Bind result (供給側起動 DSAShadowBind の場合) あるいは、
SDSM-ShSCi 状態 2 Wait for Bind result (消費側起動 DSAShadowBind の場合)
後条件: SDSM-ChT 状態 1 Idle (DSABind の場合) あるいは、
SDSM-ShCSi 状態 1 Idle (供給側起動 DSAShadowBind の場合) あるいは、
SDSM-ShSCi 状態 1 Idle (消費側起動 DSAShadowBind の場合)

S D F はセキュリティ理由のオペレーションを実行できないため directoryBindError を S D F に送信する。
DSABind の場合エラーは S C F に送信される。

16.1.29 ScfBindFailure #
16.1.30 ScfTaskRefused #
16.2 エンティティ関連エラー手順 #

17 . 詳細オペレーション手順

次の詳細手順記述には記述されていない 4,7,8 章からのオペレーションとパラメータの予定された使用は、I N C S - 3 での今後の検討課題となる。さらに詳しい情報については T T C 標準 J T - Q 1 2 2 4 の 12 章を参照のこと。

17.1 ActiveServiceFiltering 手順 #
17.2 ActivationReceivedAndAuthorized 手順 #
17.3 ActivityTest 手順 #

17.4 AddEntry 手順

17.4.1 一般記述

T T C 標準 J T - X 5 0 0 の AddEntry オペレーションは D I T への葉エントリ (オブジェクトエントリあるいはエリアエントリ) の追加を S D F に要求するために用いられる。AddEntry オペレーションの完全な記述については I T U - T 勧告 X . 5 1 1 の 11.1 節を参照。

17.4.1.1 パラメータ

I T U - T 勧告 X . 5 1 1 の 11.1.1 節及び 11.1.2 節を参照。

17.4.2 起動側エンティティ(S C F)

17.4.2.1 通常手順

S C F 前条件：

- (1) SCSM： SDF Bound あるいは、Wait for Subsequent Requests

S C F 後条件：

- (1) SCSM： SDF Bound

SCSM が Wait for Subsequent Requests 状態にあり、S D F におけるエントリの追加がサービス論理において必要である場合、内部イベント((e2)Request_to_SDF)が生じる。アプリケーションプロセスが、デリミタ(またはタイマの満了)によってオペレーションの送信を指示しない限り、SCSM は Wait for Subsequent Requests 状態に留まりオペレーションは送信されない。オペレーションは Bind アーギュメントを含んだメッセージによって S D F に送信される。SCSM は S D F からの応答を待つ。以前に S D F に対して発行された Bind オペレーションに対する応答の受信((E5)Response_from_SDF_with_Bind あるいは(E4)Bind_Error)を受けて、S C F は SDF Bind 状態または Idle 状態へ遷移する。SCSM が Idle 状態に遷移した場合、AddEntry オペレーションは破棄される。SDF_Bind 状態において AddEntry オペレーションの応答((E7)Response_from_SDF)を受けると S C F は同一の状態(SDF_Bind 状態)へ遷移する。その応答は、AddEntry オペレーションの結果、あるいはエラーであるかも知れない。

SCSM が SDF_Bind 状態にあり、S D F におけるエントリの追加がサービス論理において必要である場合、内部イベントが生じる。このイベントは(e6)Request_to_SDF と呼ばれ、全く同じ SDF_Bind 状態への遷移を引き起こし、SCSM は S D F からの応答を待つ。以前に S D F に対して発行された AddEntry オペレーションに対する応答の受信((E7)Response_from_SDF)を受けて、S C F は全く同じ SDF_Bind 状態へ遷移する。S D F からの応答は、AddEntry オペレーションの結果、あるいはエラーであるかも知れない。

17.4.2.2 エラー手順

オペレーションに関連するエラーの一般的な扱いは I T U - T 勧告 X . 5 1 1 の 11.1.4 節及び 11.1.5 節に、オペレーションエラーを通知するための T C サービスは第 4 編 18 章に記述されている。

17.4.3 応答側エンティティ(S D F)

17.4.3.1 通常手順

S D F 前条件：

- (1) SDSM： SCF Bound あるいは、Bind Pending

S D F 後条件：

- (1) SDSM： SCF Bound

S D F は Bind Pending 状態である場合、S C F からの AddEntry オペレーションの受信により(E3)Request_from_SCF という外部イベントが生ずる。S D F は Bind オペレーションの実行が成功するまでオペレーションを処理せず、同一状態に留まる。

S D F が SCF Bound 状態である場合、S C F からの AddEntry オペレーションの受信により引き起こされる(E7)Request_from_SCF という外部イベントが生ずる。S D F はオペレーションへの応答を待つ。

イベント(E7)を受信すると新たなエントリを追加する前に、S D F は以下の動作を行う。

- エントリ追加が行われる上位オブジェクトが S D F に存在することの検証

- S D F に追加するエントリが未だ存在しないことの検証
- エントリとその構成要素（属性及び値）を追加するアクセス権が十分であることの検証
- エントリがディレクトリスキーマに従っていることの検証

上述の動作が完全に行われた後、エントリが S D F のデータベースに追加される。S C F に対して空の結果が返送される。結果応答の送信は(e6)Response_to_SCF というイベントに対応する。

17.4.3.2 エラー手順

オペレーションに関連するエラーの一般的な扱いは I T U - T 勧告 X . 5 1 1 の 11.1.4 節及び 11.1.5 節に、オペレーションエラーを通知するための T C サービスは本標準の 18 章に記述されている。

17.5	AnalysedInformation 手順	#
17.6	AnalyseInformation 手順	#
17.7	ApplyCharging 手順	#
17.8	ApplyChargingReport 手順	#
17.9	AssistRequestInstructions 手順	#
17.10	AssociationReleaseRequested 手順	#
17.11	AuthorizeTermination 手順	#
17.12	CallGap 手順	#
17.13	CallInformationReport 手順	#
17.14	CallInformationRequest 手順	#
17.15	Cancel 手順	#
17.16	CancelStatusReportRequest 手順	#

17.17 chainedAddEntry 手順

17.17.1 一般記述

T T C 標準 J T - X 5 0 0 の chainedAddEntry オペレーションは、エンドユーザに代わって AddEntry オペレーションを遠隔で処理することを要求するために使用する。chainedAddEntry オペレーションの完全な記述については I T U - T 勧告 X . 5 1 1 の 12.1 節を参照。

17.17.1.1 パラメータ

I T U - T 勧告 X . 5 1 8 の 12.1 節を参照。

17.17.2 起動側エンティティ (S D F)

17.17.2.1 通常手順

S D F 前条件：

- (1) 起動側 S D F は、AddEntry オペレーションの実行要求を受信しておりエンドユーザの代わりに応答側 S D F に連鎖されるべきオペレーションを要求すること
- (2) SDSM-ChI : SDF Bound あるいは、Wait for Subsequent Requests

S D F 後条件：

- (1) SDSM-ChI : SDF Bound
- (2) AddEntry オペレーション実行要求に対する応答を、起動側 S D F で受信している。

SDSM-ChI が “ Wait for Subsequent Requests ” 状態であり、しかも応答側 S D F での処理を要求するサービスデ

ータ内エントリを追加する要求を受信している場合、内部イベント（(e2) Request_to_SDF）が生じる。アプリケーションプロセスが、デリミタ（またはタイマの満了）によってオペレーションの送信を指示しない限り、SDSM-ChIは“Wait for Subsequent Requests”状態に留まりオペレーションは送出されない。本オペレーションは、Bind アーギュメントを含んだメッセージで応答側 S D F に送出される。SDSM-ChIは、応答側 S D F からの応答を待つ。以前に S D F に対して発行された Bind オペレーションへの応答の受信（(E5) DSABind_Successful または (E4) Bind_Error）により、S C F は SDF Bound 状態または Idle 状態に遷移する。SDSM-ChI が Idle 状態に遷移した場合、AddEntry オペレーションは破棄される。SDF Bound 状態では、chainedAddEntry オペレーションの応答（(E7) Response_from_SDF）により、S C F は同じ状態（“SDF Bound”状態）に遷移する。これが chainedAddEntry オペレーションの結果、あるいはエラーであるかもしれない。この応答が発側エンドユーザに返送される。

SDSM-ChI が SDF Bound 状態であり、しかも応答側 S D F での処理を要求するサービスデータ内エントリを追加する要求を受信している場合、内部イベントが生じる。(e8) Request_to_SDF と呼ばれるこのイベントは、同じ SDF Bound 状態への遷移を引き起こし、SDSM-ChI は応答側 S D F からの応答を待つ。以前に応答側 S D F に対して発行された chainedAddEntry オペレーションに対する応答の受信（(E7) Response_from_SDF）により、起動側 S D F は同じ SDF Bound 状態への遷移を引き起こす。応答側 S D F からの応答は、chainedAddEntry オペレーションの結果、あるいはエラーであるかもしれない。この応答は発側エンドユーザに返送される。

17.17.2.2 エラー手順

オペレーションに関連するエラーの一般的な扱いは 16 章と ITU-T 勧告 X.518 の 13 章に、オペレーションエラーを通知するための T C サービスは 18 章に記述される。

17.17.3 応答側エンティティ (S D F)

17.17.3.1 通常手順

S D F 前条件：

- (1) SDSM-ChT： SDF Bound あるいは、Bind Pending

S D F 後条件：

- (1) SDSM-ChT： SDF Bound

応答側 S D F が Bind Pending 状態である場合、起動側 S D F からの chainedAddEntry オペレーションの受信により外部イベント (E3) Request_from_SDF が生じる。応答側 S D F は、dSABind オペレーションの実行が成功するまでオペレーションを処理せず、同一状態に留まる。dSABind オペレーションが失敗した場合には、本オペレーションは破棄される。dSABind オペレーションが成功した場合には、応答側 S D F によって chainedAddEntry オペレーションが処理される。

応答側 S D F が SDF Bound 状態である場合、起動側 S D F からの chainedAddEntry オペレーションの受信により外部イベント (E7) Request_from_SDF が生じる。本オペレーションは、応答側 S D F によって処理される。

応答側 S D F は、以下の 2 つの方法の 1 つで chainedAddEntry オペレーションを処理してよい。

- (1) 起動側 S D F が他網にある場合、応答側 S D F は本オペレーションを応答側 S D F と同じ網にある別の S D F と連鎖してよい。
- (2) 本オペレーションは、AddEntry 手順に記述される動作に従って処理される。

応答側 S D F が本オペレーションの処理を完了した後、本オペレーションからの結果またはエラーが起動側 S D F に返送される。本応答の送出は、イベント (e6) Response_to_SDF に相当する。

17.17.3.2 エラー処理

オペレーションに関連するエラーの一般的な扱いについては 16 章と I T U - T 勧告 X . 5 1 8 の 13 章に、オペレーションエラーを通知するための T C サービスは 18 章に記述されている。

17.18 chainedConfirmedNotificationProvided 手順 #

17.19 chainedConfirmedReportChargingInformation 手順 #

17.20 chainedEstablishChargingRecord 手順 #

17.21 chainedExecute 手順

17.21.1 一般記述

chainedExecute オペレーションは、エンドユーザに代わって Execute オペレーションを遠隔で処理することを要求するために使用する。メカニズムを連鎖する本オペレーションの完全な記述については、I T U - T 勧告 X . 5 1 8 の 12.1 節を参照。

17.21.1.1 パラメータ

I T U - T 勧告 X . 5 1 8 の 12.1 節と Execute オペレーションパラメータを参照。

17.21.2 起動側エンティティ (S D F)

17.21.2.1 正常手順

S D F 前条件：

- (1) 起動側 S D F は Execute オペレーションの実行要求を受信しており、エンドユーザの代りに応答側 S D F に連鎖されるべきオペレーションを要求すること。
- (2) SDSM-ChI： SDF Bound あるいは、Wait for Subsequent Requests

S D F 後条件：

- (1) SDSM-ChI： SDF Bound
- (2) Execute オペレーション実行要求に対する応答が、起動側 S D F から返信される。

SDSM-ChI が Wait for Subsequent Requests 状態であり、かつ、サービスデータ内にエントリを追加する要求を受信していて応答側 S D F 内の処理を要求する場合、内部イベント ((e2) Request_to_SDF) が生じる。アプリケーションプロセスがデリミタ (またはタイマ満了) によって本オペレーションの送信を指示しない限り、SDSM-ChI は Wait for Subsequent Requests 状態に留まり、本オペレーションは送出されない。本オペレーションは、Bind アーギュメントを含むメッセージで応答側 S D F に対して送出される。SDSM-ChI は、応答側 S D F からの応答を待つ。以前に S D F に対して発行された Bind オペレーションへの応答 ((E5) DSABind_Successful または (E4) Bind_Error) の受信により、S D F は SDF Bound 状態または Idle 状態に遷移する。SDSM-ChI が Idle 状態に遷移したとき、Execute オペレーションは破棄される。SDF Bound 状態では、chainedExecute オペレーション ((E7) Response_from_SDF) の応答により、S D F は同じ SDF Bound 状態に遷移する。これは chainedExecute オペレーションからの結果、あるいはエラーであるかもしれない。この応答が発側エンドユーザに返送される。

SDSM-ChI が SDF Bound 状態であり、かつ、サービスデータ内にエントリを追加する要求を受信していて応答側 S D F 内の処理を要求する場合、内部イベントが生じる。このイベントは (e8) Request_to_SDF と呼ばれ、これにより同じ SDF Bound 状態に遷移し、SDSM-ChI は応答側 S D F からの応答を待つ。以前に応答側 S D F に対して発行された chainedExecute オペレーションに対する応答 ((E7) Response_from_SDF) の受信により、起動側 S D F は同じ SDF Bound 状態に遷移する。応答側 S D F からの応答は、chainedExecute オペレーションからの結果、あるいはエラーであるかもしれない。この応答が発側エンドユーザに返送される。

17.21.2.2 エラー処理

オペレーションに関連するエラーの一般的な扱いは 16 章と I T U - T 勧告 X . 5 1 8 の 13 節に、オペレーションエラーを通知するための T C サービスは 18 章に記述される。

17.21.3 応答側エンティティ (S D F)

17.21.3.1 正常手順

S D F 前条件 :

- (1) SDSM-ChT : SDF Bound あるいは、Bind Pending

S D F 後条件 :

- (1) SDSM-ChT : SDF Bound

応答側 S D F が Bind Pending 状態である場合、起動側 S D F から chainedExecute オペレーションを受信することにより外部イベント (E3) Request_from_SDF が生じる。応答側 S D F は、dSABind オペレーションが成功するまでは本オペレーションの処理を続行しない。同じ状態であり続ける。dSABind オペレーションが失敗した場合には、本オペレーションは破棄される。dSABind オペレーションが成功した場合には、応答側 S D F によって chainedExecute オペレーションが処理される。

応答側 S D F が SDF Bound 状態である場合、起動側 S D F から chainedExecute オペレーションを受信することにより外部イベント (E7) Request_from_SDF が生じる。このオペレーションは、応答側 S D F によって処理される。

応答側 S D F は、以下の 2 つのうちの 1 つの方法で chainedExecute オペレーションを処理してよい。

- (1) 起動側 S D F が他網に存在する場合、応答側 S D F は自網内の他の S D F にそのオペレーションを連鎖することができる。
- (2) 本オペレーションは、Execute 手順に記述される動作に従って処理される。

応答側 S D F が本オペレーションの処理を完了した後、本オペレーションから生じた何らかの結果、または、エラーが起動側 S D F に返送される。この応答の送出手は、イベント (e6) Response_to_SDF に相当する。

17.21.3.2 エラー処理

オペレーションに関連するエラーの一般的な扱いは 16 章と I T U - T 勧告 X . 5 1 8 の 13 節に、オペレーションエラーを通知するための T C サービスは 18 章に記述される。

17.22 chainedHandlingInformationRequest 手順 #

17.23 chainedHandlingInformationResult 手順 #

17.24 chainedModifyEntry 手順

17.24.1 一般記述

T T C 標準 J T - X 5 0 0 の chainedModifyEntry オペレーションは、エンドユーザに代って ModifyEntry オペレーションを遠隔で処理することを要求するために使用する。chainedModifyEntry オペレーションの完全な記述については、I T U - T 勧告 X . 5 1 8 の 12.1 節を参照。

17.24.1.1 パラメータ

I T U - T 勧告 X . 5 1 8 の 12.1 節を参照。

17.24.2 起動側エンティティ (S D F)

17.24.2.1 正常手順

S D F 前条件 :

- (1) 起動側 S D F は、ModifyEntry オペレーションの実行要求を受信しており、エンドユーザの代りに応答側 S D F に連鎖すべきオペレーションを要求すること。
- (2) SDSM-ChI : SDF Bound または Wait for Subsequent Requests

S D F 後条件 :

- (1) SDSM-ChI : SDF Bound
- (2) ModifyEntry オペレーションの実行要求に対する応答が、起動側 S D F から返信される。

SDSM-ChI が Wait for Subsequent Requests 状態であり、かつ、サービスデータ内でエントリを更新する要求を受信していて応答側 S D F 内の処理を要求する場合、内部イベント ((e2) Request_to_SDF) が生じる。アプリケーションプロセスがデリミタ (またはタイマ満了) によって本オペレーションの送信を指示しない限り、SDSM-ChI は Wait for Subsequent Requests 状態に留まり、本オペレーションは送出されない。本オペレーションは、Bind アーギュメントを含むメッセージで応答側 S D F に対して送出される。SDSM-ChI は、応答側 S D F からの応答を待つ。以前に S D F に対して発行された Bind オペレーションへの応答 ((E5) DSABind_Successful または (E4) Bind_Error) の受信により、S D F は SDF Bound 状態または Idle 状態に遷移する。SDSM-ChI が Idle 状態に遷移した場合、ModifyEntry オペレーションは破棄される。SDF Bound 状態では、chainedModifyEntry オペレーション ((E7) Response_from_SDF) の応答により、S D F は同じ SDF Bound 状態に遷移する。これは chainedModifyEntry オペレーションからの結果、あるいはエラーであるかもしれない。この応答が発側エンドユーザに返送される。

SDSM-ChI が SDF Bound 状態であり、かつ、サービスデータ内のエントリを更新する要求を受信していて応答側 S D F 内の処理を要求する場合、内部イベントが生じる。このイベントは (e8) Request_to_SDF と呼ばれ、これにより同じ SDF Bound 状態に遷移し、SDSM-ChI は応答側 S D F からの応答を待つ。以前に応答側 S D F に対して発行された chainedModifyEntry オペレーションに対する応答 ((E7) Response_from_SDF) の受信により、起動側 S D F は同じ SDF Bound 状態に遷移する。応答側 S D F からの応答は、chainedModifyEntry オペレーションからの結果、あるいはエラーであるかもしれない。この応答が発側エンドユーザに返送される。

17.24.2.2 エラー処理

オペレーションに関連するエラーの一般的な扱いは 16 章と I T U - T 勧告 X . 5 1 8 の 13 章に、オペレーションエラーを通知するための T C サービスは 18 章に記述される。

17.24.3 応答側エンティティ (S D F)

17.24.3.1 正常手順

S D F 前条件 :

- (1) SDSM-ChT : SDF Bound あるいは、Bind Pending

S D F 後条件 :

- (1) SDSM-ChT : SDF Bound

応答側 S D F の状態が Bind Pending である場合は、起動側 S D F から chainedModifyEntry オペレーションを受信することにより、外部イベント (E3) Request_from_SDF が発生する。応答側 S D F は、dSABind オペレーションが正常に実行されるまでは、本オペレーションの実行を進行しない。同一の状態が保たれる。もし dSABind オペレーションが失敗した場合は、本オペレーションは廃棄される。もし dSABind オペレーションが成功した場合は、応答側 S D F によって chainedModifyEntry オペレーションの処理が行われる。

応答側 S D F の状態が SCF Bound である場合は、起動側 S D F から chainedModifyEntry オペレーションを受信することにより、外部イベント(E7) Request_from_SDF が発生する。このオペレーションは応答側 S D F によって処理される。

応答側 S D F は、chainedModifyEntry オペレーションを、次の 2 つのうちいずれか一方の方法で処理することができる。

- (1) 起動側 S D F が他網にある場合、応答側 S D F は自網内の他の S D F にそのオペレーションを連鎖することができる。
- (2) 本オペレーションは、ModifyEntry 手順で記述された動作に従って処理される。

応答側 S D F が本オペレーションの処理を終了した後、本オペレーションからの任意の結果またはエラーが、起動側 S D F へ返される。この応答を送信することにより、イベント(e6) Response_to_SDF が発生する。

17.24.3.2 エラー処理

オペレーションに関連するエラーの一般的な扱いは 16 章と I T U - T 勧告 X . 5 1 8 の 13 章に、オペレーションエラーを通知するための T C サービスは 18 章に記述される。

17.25	chainedNetworkCapability 手順	#
17.26	chainedNotificationProvided 手順	#
17.27	chainedReportChargingInformation 手順	#
17.28	chainedProvideUserInfoInformation 手順	#

17.29 chainedRemoveEntry 手順

17.29.1 一般記述

T T C 標準 J T - X 5 0 0 の chainedRemoveEntry オペレーションは、エンドユーザに代わって RemoveEntry オペレーションの遠隔処理を要求するために使用される。chainedRemoveEntry オペレーションの完全な記述については、I T U - T 勧告 X . 5 1 8 の 12.1 節を参照のこと。

17.29.1.1 パラメータ

I T U - T 勧告 X . 5 1 8 の 12.1 節を参照のこと。

17.29.2 起動側エンティティ (S D F)

17.29.2.1 正常手順

S D F 前条件 :

- (1) 起動側 S D F は、RemoveEntry オペレーションの実行要求を受信しており、エンドユーザの代わりに応答側 S D F に連鎖されるべきオペレーションを要求すること。
- (2) SDSM-ChI : SDF Bound あるいは、Wait for SubsequentRequests

S D F 後条件 :

- (1) SDSM-ChI : SDF Bound
- (2) 起動側 S D F が、RemoveEntry オペレーションの実行要求に対する応答を受信した。

SDSM-ChI の状態が Wait for Subsequent Requests で、かつサービスデータからのエントリ削除要求を受信し、応答側 S D F での処理が必要とされる場合は、内部イベント(e2) Request_to_SDF が発生する。アプリケーションプロセスが、本オペレーションが送信されるべきことをデリミタ(またはタイマの満了) によって指示しないうちは、SDSM-ChI の状態は Wait for Subsequent Requests に留まり、本オペレーションは送信されない。本オペレ-

ションは、Bind アーギュメントを含むメッセージにより、応答側 S D F へ送信される。SDSM-ChI は応答側 S D F からの応答を待つ。先に応答側 S D F に対して発行された Bind オペレーションに対する応答((E5) DSABind_Successful または(E4) Bind_Error)を受信することにより、S D F の状態は SDF Bound あるいは Idle に遷移する。SDSM-ChI の状態が Idle に遷移した場合は、RemoveEntry オペレーションは廃棄される。状態 SDF Bound においては、chainedRemoveEntry オペレーションに対する応答((E7) Response_from_SDF)により、S C F は同一の状態(SDF Bound)に遷移する。この応答は、chainedRemoveEntry オペレーションの結果がエラーのいずれかである。この応答は発側のエンドユーザに返送される。

SDSM-ChI の状態が SDF Bound で、かつサービスデータからのエントリ削除要求を受信し、応答側 S D F での処理が必要とされる場合は、内部イベントが発生する。このイベントは(e8) Request_to_SDF であり、これにより同一の状態 SDF Bound に遷移し、また SDSM-ChI は応答側 S D F からの応答を待つ。先に応答側 S D F に対して発行された chainedRemoveEntry オペレーションに対する応答((E7) Response_from_SDF)を受信することにより、起動側 S D F は同一の状態 SDF Bound に遷移する。S D F からの応答は、chainedRemoveEntry オペレーションの結果あるいはエラーのいずれかである。この応答は発側のエンドユーザに返送される。

17.29.2.2 エラー処理

オペレーションに関連するエラーの一般的な扱いは 16 章と I T U - T 勧告 X . 5 1 8 の 13 章に、オペレーションエラーを通知するための T C サービスは 18 章に記述される。

17.29.3 応答側エンティティ (S D F)

17.29.3.1 正常手順

S D F 前条件 :

- (1) SDSM-ChT : SDF Bound あるいは、Bind Pending

S D F 後条件 :

- (1) SDSM-ChT : SDF Bound

応答側 S D F の状態が Bind Pending である場合は、起動側 S D F から chainedRemoveEntry オペレーションを受信することにより、外部イベント(E3) Request_from_SDF が発生する。応答側 S D F は、dSABind オペレーションが正常に実行されるまでは、本オペレーションの実行を進行しない。同一の状態が保たれる。もし dSABind オペレーションが失敗した場合は、本オペレーションは廃棄される。もし dSABind オペレーションが成功した場合は、応答側 S D F によって chainedRemoveEntry オペレーションの処理が行われる。

応答側 S D F の状態が SDF Bound である場合は、起動側 S D F から chainedRemoveEntry オペレーションを受信することにより、外部イベント(E7) Request_from_SDF が発生する。このオペレーションは応答側 S D F によって処理される。

応答側 S D F は、 chainedRemoveEntry オペレーションを、次の 2 つのうちいずれか一方の方法で処理することができる。

- (1) もし起動側 S D F が他網にある場合は、応答側 S D F は、自網内の他の S D F に本オペレーションを連鎖し、応答側 S D F とすることができる。
- (2) 本オペレーションは、RemoveEntry 手順で記述された動作に従って処理される。

応答側 S D F が本オペレーションの処理を終了した後、本オペレーションからの任意の結果またはエラーが、起動側 S D F へ返される。この応答を送信することにより、イベント(e6) Response_to_SDF が発生する。

17.29.3.2 エラー処理

オペレーションに関連するエラーの一般的な扱いは 16 章と I T U - T 勧告 X . 5 1 8 の 13 章に、オペレーシ

ョンエラーを通知するためのTCサービスは18章に記述される。

17.30 chainedRequestNotification 手順

#

17.31 chainedSearch 手順

17.31.1 一般記述

TTT標準JT-X500のchainedSearchオペレーションはエンドユーザに代わってSearchオペレーションを遠隔で処理するために使用する。chainedSearchオペレーションの完全な記述についてはITU-T勧告X.518の12.1節を参照

17.31.1.1 パラメータ

ITU-T勧告X.518の12.1節を参照。

17.31.2 起動側エンティティ(SDF)

17.31.2.1 正常手順

SDF前条件:

- (1) 起動側SDFは、Searchオペレーションの実行要求を受信しており、エンドユーザの代わりに応答側SDFに連鎖されるべきオペレーションを要求すること。
- (2) SDSM-ChI: SDF Bound あるいは、Wait for Subsequent Requests

SDF後条件:

- (1) SDSM-ChI: SDF Bound
- (2) Searchオペレーション実行要求に対する応答を起動側SDFで受信している。

SDSM-ChIがWait for Subsequent Requestの状態であつた応答側SDFにおいて処理が必要とされるサービスデータのエントリの探索要求を受信している時に内部イベント(e2)Request_to_SDFが発生する。アプリケーションプロセスがデリミタ(またはタイマーの満了)によって本オペレーションの送出手示しない限り、SDSM-ChIはWait for Subsequent Requestsの状態に留まりオペレーションは送信されない。本オペレーションはBindアークギュメントを含んだメッセージによって応答側SDFに送信される。SDSM-ChIは応答側SDFからの応答を待つ。以前にSDFに対して発行されたBindオペレーションに対する応答((E5)DSABind_Successful あるいは(E4)Bind_Error)の受信により、SDFはSDF Bound あるいはIdleの状態に遷移する。SDSM-ChIがIdle状態に遷移したときにSearchオペレーションは破棄される。SDF Bound状態においてchainedSearchオペレーションの応答((E5)DSABind_Successful あるいは(E4)Bind_Error)を受けるとSDFは同一の状態(SDF Bound)に遷移する。その応答はchainedSearchオペレーションの結果あるいはエラーであるかもしれない。この応答は発側エンドユーザに返送される。

SDSM-ChIがSDF Boundの状態であつた応答側SDFにおいて処理が必要とされるサービスデータのエントリの探索要求を受信している時に内部イベントが発生する。このイベントは(e8)Request_to_SDFと呼ばれ同じ状態のSDF Boundに遷移し、SDSM-ChIは応答側SDFからの応答を待つ。以前に応答側SDFに対して発行されたchainedSearchオペレーションに対する応答(E7)Response_from_SDFの受信により、起動側SDFは同じ状態のSDF Boundに遷移する。応答側SDFからの応答は、chainedSearchオペレーションの結果あるいはエラーであるかもしれない。この応答は発側エンドユーザに返送される。

17.31.2.2 エラー処理

オペレーションに関連するエラーの一般的な扱いは、16章およびITU-T勧告X.518の13章に、オペ

レーションエラーを通知するためのTCサービスは18章に記述されている。

17.31.3 応答側エンティティ (SDF)

17.31.3.1 正常手順

SDF前条件:

- (1) SDSM-ChT: SDF Bound あるいは、Bind Pending

SDF後条件:

- (1) SDSM-ChT: SDF Bound

応答側SDFがBind Pending状態である場合には、起動側SDFからのchainedSearchオペレーションを受信することにより、外部イベント(E3)Request_from_SDFが生じる。応答側SDFはdSABindオペレーションの実行が成功するまでオペレーションを処理せず、同一の状態に留まる。もしdSABindが失敗した場合には、オペレーションは破棄される。もしdSABindオペレーションが成功した場合には、chainedSearchオペレーションは、応答側SDFによって処理される。

応答側SDFがSDF Bound状態である場合には、起動側SDFからのchainedSearchオペレーションを受信することにより、外部イベント(E7)Request_from_SDFが生じる。このオペレーションは応答側SDFによって処理される。

応答側SDFは、以下のどちらかの方法でchainedSearchオペレーションを処理することができる。

- (1) 起動側SDFが他網に存在する場合、応答側SDFは自網内の他のSDFにそのオペレーションを連鎖することができる。
- (2) オペレーションはSearch手順で記述されている動作によって処理される。

応答側SDFが、オペレーション処理を終了した後、オペレーションから何らかの結果またはエラーが起動側SDFへ返送される。この応答の送信はイベント(e6)Response_to_SDFに相当する。

17.31.3.2 エラー処理

オペレーションに関連するエラーの一般的な扱いは、16章およびITU-T勧告X.518の13章に、オペレーションエラーを通知するためのTCサービスは18章に記述されている。

17.32	collectedInformation 手順	#
17.33	collectInformation 手順	#
17.34	componentReceived 手順	#
17.35	confirmedNotificationProvided 手順	#
17.36	confirmedReportChargingInformation 手順	#
17.37	connect 手順	#
17.38	connectToResouce 手順	#
17.39	continue 手順	#
17.40	continueWithArgument 手順	#

17.41 CoordinateShadowUpdate 手順

17.41.1 一般記述

TTTC標準JT-X500のシャドウイング(shadowing)オペレーションにより2つのSDF間で情報がコピーされる。シャドウイングオペレーションはまた、このコピーされた情報を保持するためにも使用される。2つのSDF間の個々の各シャドウイング合意において、一方のSDFをコピーされた情報の供給側と呼び、他方

の S D F を消費側と呼ぶ。

DSAShadowBind オペレーションと DSAShadowUnbind オペレーションは、コピーを提供するある期間の始めと終わりに、協調して動作する S D F によって使用される。CoordinateShadowUpdate オペレーションは、シャドウの供給側が更新を送ることを意図するシャドウイング合意を指示するために使用される。requestShadowUpdate オペレーションは、シャドウ消費側がシャドウ供給側からの更新を要求するために使用される。UpdateShadow オペレーションは、シャドウ供給側がシャドウ消費側へコピーされたデータを送るために使用される。このオペレーションは、CoordinateShadowUpdate オペレーションあるいは requestShadowUpdate オペレーションが最初に処理されていなければならない。シャドウイングオペレーションの完全な記述については I T U - T 勧告 X . 5 2 5 を参照。

17.41.1.1 パラメータ

CoordinateShadowUpdate オペレーションについては I T U - T 勧告 X . 5 2 5 の 11.1 節を参照。

17.41.2 供給側エンティティ (S D F)

17.41.2.1 正常手順

17.41.2.1.1 自身の送信した CoordinateShadowUpdate

S D F 前条件 :

- (1) SDSM-ShSSi : SDF Bound

S D F 後条件 :

- (1) SDSM-ShSSi : Wait for Update (CoordinateShadowUpdate が成功した場合)
- (2) SDSM-ShSSi : SDF Bound (CoordinateShadowUpdate が失敗した場合)

SDSM-ShSSi が SDF Bound の状態にあり、シャドウの調整が必要である場合、内部イベントが生じる。このイベントは、(e16)Shadow_Coordinate_to_Consumer と呼ばれ、Wait for Coordination Result の状態への遷移を引き起こし、消費側 S D F へオペレーションが送信される。SDSM-ShSSi は消費側 S D F からの応答を待つ。CoordinateShadowUpdate オペレーションが成功した場合には、以前に消費側 S D F に対して発行された CoordinateShadowUpdate オペレーションに対する応答(E18)Shadow_Coordinate_Confirmed の受信により、消費側 S D F は Wait for Update の状態に遷移する。その他の場合には、SDSM-ShSSi はエラー(E19)Coordinate_Failure の受信により SDF Bound の状態へ戻る。

17.41.2.1.2 DSAShadowBind とともに送信した CoordinateShadowUpdate

S D F 前条件 :

- (1) SDSM-ShSSi : Wait for Subsequent Requests

S D F 後条件 :

- (1) SDSM-ShSSi : Wait for Update (CoordinateShadowUpdate が成功した場合)
- (2) SDSM-ShSSi : SDF Bound (CoordinateShadowUpdate が失敗した場合)

SDSM-ShSSi が Wait for Subsequent Requests の状態にあり、シャドウの調整が必要である場合、内部イベントが生じる。このイベントは、(e2)Shadow_Coordinate_to_Consumer と呼ばれ、Bind with Coordinate Shadow の状態へ遷移する。デリミタの受信により内部イベント(e5)Send_Bind_with_CoordShadow が発生し、Bind with Coordinate Shadow only 状態へ遷移する。そしてオペレーションが消費側 S D F へ送信される。SDSM-ShSSi は、消費側からの応答を待っている。以前に発行した DSAShadowBind への応答(E7)SDF_Bind_Success の受信により Wait for Coordination Result の状態への遷移が引き起こる。SDSM-ShSSi は、以前に消費側 S D F に対して発行された CoordinateShadowUpdate オペレーションの消費側からの応答を待っている。CoordinateShadowUpdate オペレーシ

ヨンが成功した場合には、イベント(E18)Shadow_Coordinate_Confirmed は、Wait for Update の状態へ遷移する。その他の場合には、SDSM-ShSSi はエラー(E19)Coordinate_Failure の受信により SDF Bound の状態へ戻る。

17.41.2.1.3 DSAShadowBind および UpdateShadow とともに送信した CoordinateShadowUpdate

S D F 前条件 :

- (1) SDSM-ShSSi : Wait for Subsequent Requests

S D F 後条件 :

- (1) SDSM-ShSSi : Wait for Update (CoordinateShadowUpdate が成功した場合)
- (2) SDSM-ShSSi : SDF Bound (CoordinateShadowUpdate が失敗した場合)

SDSM-ShSSi が Wait for Subsequent Requests の状態にあり、シャドウの調整が必要である場合、内部イベントが生じる。このイベントは、(e2)Shadow_Coordinate_to_Consumer と呼ばれ、Bind with Coordinate Shadow の状態へ遷移する。後続の UpdateShadow オペレーションの受信により内部イベント(e4)Update_to_Consumer が発生し、Bind with CoordinateShadow and Update 状態に遷移する。そしてオペレーションが消費側 S D F へ送信される。SDSM-ShSSi は消費側からの応答を待っている。以前に発行した DSAShadowBind への応答(E9)SDF_Bind_Success の受信により、Bound with Coordinate Shadow Sent の状態へ遷移する。SDSM-ShSSi は、以前に消費側 S D F に対して発行された CoordinateShadowUpdate オペレーションの消費側からの応答を待っている。CoordinateShadowUpdate オペレーションが成功した場合には、イベント(E10)Shadow_Coordinate_Confirmed により、Wait for Update Confirmation の状態へ遷移する。その他の場合には、SDSM-ShSSi はエラー(E11)Coordinate_Failure の受信により SDF Bound の状態へ戻る。

17.41.2.2 エラー処理

シャドウイングオペレーションに関連するエラーの一般的な扱いは I T U - T 勧告 X . 5 2 5 の 12 章に、オペレーションエラーを通知するための T C サービスは 18.1 節に記述されている。

17.41.3 消費側エンティティ (S D F)

17.41.3.1 正常手順

17.41.3.1.1 自身の受信した CoordinateShadowUpdate

S D F 前条件 :

- (1) SDSM-ShCSi : SDF Bound

S D F 後条件 :

- (1) SDSM-ShCSi : Wait for Update (coordinateShadowUpdate が成功した場合)
- (2) SDSM-ShCSi : SDF Bound (coordinateShadowUpdate が失敗した場合)

消費側 S D F は最初 SDF Bound 状態にある。供給側 S D F からの coordinateShadowUpdate オペレーション受信により引き起こされる外部イベント(E7)Shadow_Coordinate_from_Supplier の受信後、Wait for Coordination Result 状態へ遷移する。消費側 S D F は coordinateShadowUpdate アーギュメントの内容に従って coordinateShadowUpdate オペレーションを実行する。消費側 S D F が coordinateShadowUpdate オペレーションを完了させると、結果応答あるいはエラーが供給側 S D F に対して返される。消費側 S D F は coordinateShadowUpdate が失敗した場合には SDF Bound 状態に戻り、成功した場合には Wait for Update 状態へ遷移する。

17.41.3.1.2 DSAShadowBind または DSAShadowBind と UpdateShadow とともに受信した CoordinateShadowUpdate

S D F 前条件 :

- (1) SDSM-ShCSi : Wait for Bind Result

SDF 後条件 :

- (1) SDSM-ShCSi : Wait for Update (coordinateShadowUpdate が成功した場合)
(2) SDSM-ShCSi : SDF Bound (coordinateShadowUpdate が失敗した場合)

消費側 SDF は最初 Wait for Bind Result 状態にあり、DSAShadowBind オペレーション及び、同時に受信すべきその他のオペレーションを待っている。CoordinateShadowUpdate オペレーションの受信により引き起こされる (E3)Request_from_Supplier という外部イベントによって、同じ状態へ遷移する。さらに UpdateShadow オペレーションを受信した場合も同じ状態へ遷移する。消費側 SDF は DSAShadowBind オペレーションを実行し (e5)SDF_Bind_Success という内部イベントによって、SDF Bound 状態に遷移する。既に CoordinateShadowUpdate オペレーションを受信している場合は、(E7)Shadow_Coordinate_from_Supplier という外部イベントによって Wait for Coordination Result 状態へ遷移する。その後消費側 SDF は coordinateShadowUpdate アーギュメントの内容に従って coordinateShadowUpdate オペレーションを実行する。消費側 SDF が coordinateShadowUpdate オペレーションを完了させると、結果応答あるいはエラーが供給側 SDF に対して返される。消費側 SDF は coordinateShadowUpdate が失敗した場合には SDF Bound 状態に戻り、成功した場合には Wait for Update 状態へ遷移する。

17.41.3.2 エラー処理

シャドウイングオペレーションに関連するエラーの一般的な扱いはITU-T勧告X.525の12章に、オペレーションエラーを通知するためのTCサービスは18.1節に記述されている。

17.42 CreateCallSegmentAssociation 手順

#

17.43 in-directoryBind 手順

17.43.1 一般記述

TTT標準JT-X500のdirectoryBindオペレーションは、SCFがエンドユーザに代わってSCFとSDF間の認証されたアソシエーションを確立するために使用され、エンドユーザの認証情報があれば配送する。DirectoryBindオペレーションの完全な記述についてはITU-T勧告X.511の8.1節を参照。

17.43.1.1 パラメータ

ITU-T勧告X.511の8.1.2及び8.1.3節を参照。

17.43.2 起動側エンティティ(SCF)

17.43.2.1 正常手順

SCF 前条件 :

- (1) SCSM-SDF : Idle

SCF 後条件 :

- (1) SCSM-SDF : SDF Bound(directoryBind が成功した場合)
(2) SCSM-SDF : Idle(directoryBind が失敗した場合)

SCSM-SDF が Idle 状態にあり、SDF への問い合わせがサービス論理において必要である場合、内部イベントが生じる。このイベントは(e1)Bind_Request と呼ばれ、Wait for Subsequent Requests 状態への遷移を引き起こし、他のオペレーションは待たされる。アプリケーションプロセスがデリミタによって Bind の送信を指示しない限

り、SCSM-SDFは Wait for Subsequent Requests 状態に留まり、オペレーションは送信されない。デリミタの受信により 内部イベント(e3)Request_to_SDF_with_Bind が発生し Wait for Bind Result 状態へ遷移する。オペレーションはS D Fに送信される。SCSM-SDFはS D Fからの応答を待つ。 Bind オペレーションが成功した場合は、以前にS D Fに対して発行された(E5)Response_from_SDF_with_Bindにより、S C Fの状態はSDF Bound という状態へ遷移する。その他の場合は、SCSM-SDFはエラー(E4)Bind_Errorの受信により Idle 状態へ戻る。

17.43.2.2 エラー処理

オペレーションに関連するエラーの一般的な扱いは16章とITU-T勧告X.511の8.1.4節に、オペレーションエラーを通知するためのTCサービスは18章に記述されている。

17.43.3 応答側エンティティ(S D F)

17.43.3.1 正常手順

S D F 前条件：

- (1) SDSM-SCF： Idle

S D F 後条件：

- (1) SDSM-SCF： SCF Bound(directoryBind が成功した場合)
- (2) SDSM-SCF： Idle ” (directoryBind が失敗した場合)

SDSM-SCFは最初 Idle 状態にある。S C Fからの directoryBind オペレーションの受信により生じる 外部イベント(E1)Bind_from_SCFの受信後、Bind Pending 状態へ遷移する。S D Fは directoryBind アーギュメントの内容に従って Bind オペレーションを実行する。S D Fが directoryBind オペレーションを完了させると、結果応答あるいはエラーがS C Fに対して返される。S D Fは Bind が失敗した場合には Idle 状態に戻り、Bind が成功した場合には SCF Bound 状態へ遷移する。Bind 要求が成功した場合、返送される結果応答は、DirectoryBindResult の資格証明(Credential)から構成される。これらの資格証明によりユーザはディレクトリを識別することができる。これにより(ディレクトリサービスを直接提供する)D S Aの識別情報をD U Aに伝達することができる。資格証明は、ユーザに提供されたのと同じ形式でなければならない。

17.43.3.2 エラー処理

オペレーションに関連するエラーの一般的な扱いは16章とITU-T勧告X.511の8.1.4節に、オペレーションエラーを通知するためのTCサービスは18章に記述されている。

17.44 in-directoryUnbind 手順

17.44.1 一般記述

TTTC標準JT-X500の Unbind オペレーションは、S D Fがエンドユーザに代わってS C FとS D F間の認証されたアソシエーションを終了させるために使用される。Unbind オペレーションの完全な記述については、ITU-T勧告X.511の8.1節を参照。

17.44.1.1 パラメータ

ITU-T勧告X.511の8.1.2節および8.1.3節を参照。

17.44.2 起動側エンティティ(S C F)

17.44.2.1 正常手順

S C F 前条件 :

- (1) SCSM : SDF Bound

S C F 後条件 :

- (2) SCSM : Idle

SCSM は S D F ディレクトリに対して既に Bind オペレーションを開始しており、SDF Bound 状態にある。サービス論理が認証された S D F アクセスが終了すべきことを決定する。SCSM は Unbind オペレーション (e8)Unbind_request) を発行し、Idle 状態への遷移が生ずる。

17.44.2.2 エラー処理

Unbind オペレーションは、オペレーションに関連するエラーを持たない。

17.44.3 応答側エンティティ(S D F)

17.44.3.1 正常手順

S D F 前条件 :

- (1) SDSM : SCF Bound

S D F 後条件 :

- (1) SDSM : Idle

Bind オペレーションが既に発行され、SDSM は SCF Bound 状態で S C F からの要求を待っているか、またはオペレーションを実行している。Unbind オペレーションを受信すると、(E5)Unbind_from_SCFにより Idle 状態に遷移する。

17.44.3.2 エラー処理

Unbind オペレーションは、オペレーションに関連するエラーを持たない。

17.45 DisconnectForwardConnection 手順

#

17.46 DisconnectForwardConnectionWithArgument 手順

#

17.47 DisconnectLog 手順

#

17.48 dSABind 手順

17.48.1 一般記述

T T C 標準 J T - X 5 0 0 の dSABind オペレーションは、起動側 S D F がエンドユーザに代わってオペレーションの分散処理を可能とするために起動側 S D F と応答側 S D F 間の認証されたアソシエーションを確立するために使用され、エンドユーザの認証情報がある場合には配送する。dSABind オペレーションの完全な記述については、I T U - T 勧告 X . 5 1 8 の 11.1 節を参照。分散オペレーション手順の完全な記述については、I T U - T 勧告 X . 5 1 8 を参照。

17.48.1.1 パラメータ

I T U - T 勧告 X . 5 1 8 の 11.1 節を参照。

17.48.2 起動側エンティティ (S D F)

17.48.2.1 正常手順

S D F 前条件 :

- (1) エンドユーザからデータアクセスオペレーションのための要求が到達し、それは処理のためにリモート S D F に対して連鎖される必要があり、発側エンドユーザのためには S D F 間に連鎖アソシエーションが無い
- (2) SDSM-ChI : Idle

S D F 後条件 :

- (1) SDSM-ChI : SDF Bound(dSABind が成功した場合)
- (2) SDSM-ChI : Idle(dSABind が失敗した場合)
- (3) エンドユーザのために起動側 S D F と応答側 S D F 間に連鎖アソシエーションがある

SDSM-ChI が Idle 状態にあり、別の S D F への問い合わせがサービス論理において必要である場合、内部イベントが生じる。このイベントは(e1)DSABind_to_SDF と呼ばれ、Wait for Subsequent Requests 状態への遷移を引き起こし、他のオペレーションは待たされる。アプリケーションプロセスがデリミタによって結合の送信を指示しない限り、SDSM-ChI は Wait for Subsequent Requests 状態に留まり、オペレーションは送信されない。デリミタが受信されると(e3)Send_Bind_with_Requests という内部イベントを経て Wait for Bind Result 状態へ遷移する。オペレーションは応答側 S D F に送信される。SDSM-ChI は応答側 S D F からの応答を待つ。Bind オペレーションが成功した場合は、以前に応答側 S D F に対して発行された Bind オペレーションに対する応答の受信 (E5)DSABind_Successful) により、起動側 S D F の状態は SDF Bound という状態へ遷移する。その他の場合は、SDSM-ChI はエラーの受信 (E4)DSABind_Error) により Idle 状態へ戻る。

17.48.2.2 エラー処理

オペレーションに関連するエラーの一般的な扱いは、16 章と I T U - T 勧告 X . 5 1 8 の 13 章に、オペレーションエラーを通知するための T C サービスは 18 章に記述されている。

17.48.3 応答側エンティティ (S D F)

17.48.3.1 正常手順

S D F 前条件 :

- (1) SDSM-ChT : idle

S D F 後条件 :

- (1) SDSM-ChT : SDF Bound(dSABind が成功した場合)
- (2) SDSM-ChT : idle(dSABind が失敗した場合)

SDSM-ChT は最初 idle 状態にある。起動側 S D F からの dSABind オペレーションの受信により引き起こされた外部イベント(E1)DSABind_from_SDF を受信後、Bind Pending 状態へ遷移する。応答側 S D F は、dSABind アーギュメントの内容に従い、この Bind オペレーションを実行する。S D F が dSABind オペレーションの処理を終了すると、結果あるいはエラー指示が起動側 S D F に返送される。応答側 S D F は、Bind が失敗した場合には idle 状態に戻り、Bind が成功した場合には SDF Bound 状態に遷移する。Bind 要求が成功した場合、返送される結果応答は、dSABindResult の資格証明から構成される。これらの資格証明はユーザがディレクトリを特定できるようにする。これにより (ディレクトリサービスを直接提供する) 応答側 S D F の識別情報を起動側 S D F に伝達することができる。資格証明は、ユーザに提供されたのと同じ形式でなければならない。

17.48.3.2 エラー処理

オペレーションに関連するエラーの一般的な扱いは、16章及びITU-T勧告X.518の13章、オペレーションエラーを通知するためのTCサービスは18章に記述されている。

17.49 DSAShadowBind 手順

17.49.1 一般記述

TTT標準JT-X500のシャドウイングオペレーションにより、2つのSDF間で情報がコピーされる。シャドウイングオペレーションはコピーされた情報を維持するためにも用いられる。2つのSDF間の個々のシャドウイング合意において、一方のSDFはコピーされた情報の供給側となり、他方のSDFは消費側となる。

DSAShadowBind オペレーションとDSAShadowUnbind オペレーションは、コピーを供給するある期間のはじめと終わりに、協調して動作するSDFによって使用される。coordinateShadowUpdate オペレーションは、シャドウ供給側が更新を意図する対象のシャドウイング合意を示すために使用される。requestShadowUpdate オペレーションは、シャドウ消費側がシャドウ供給側からの更新を要求するために使用される。updateShadow オペレーションは、シャドウ供給側がコピーされるデータをシャドウ消費側に送信するために起動される。このオペレーションより前に、coordinateShadowUpdate あるいは requestShadowUpdate のオペレーションが行われなければならない。シャドウイングオペレーションの完全な記述についてはITU-T勧告X.525を参照。

17.49.1.1 パラメータ

DSAShadowBind オペレーションのパラメータは、本標準に規定される in-DirectoryBind オペレーションのパラメータと同様である。

17.49.2 供給側エンティティ (SDF)

17.49.2.1 正常手順

17.49.2.1.1 供給側起動による DSAShadowBind

17.49.2.1.1.1 自身の送信した DSAShadowBind

SDF 前条件 :

- (1) SDSM-ShSSi : Idle

SDF 後条件 :

- (1) SDSM-ShSSi : SDF Bound (DSAShadowBind が成功した場合)
- (2) SDSM-ShSSi : Idle (DSAShadowBind が失敗した場合)

SDSM-ShSSi が Idle 状態にあり、コピーの供給が必要である場合、内部イベントが生じる。このイベントは (e1)Bind_to_Consumer と呼ばれ、Wait for Subsequent Requests 状態への遷移を引き起こし、他のオペレーションは待たされる。DSAShadowBind オペレーションと同時に送信すべき後続のオペレーションを許容するため、DSAShadowBind は遅延する。アプリケーションプロセスがデリミタによってDSAShadowBindの送信を指示しない限り、SDSM-ShSSiはWait for Subsequent Requests状態に留まり、オペレーションは送信されない。デリミタが受信されると、(e3)Send_Bindという内部イベントを経て、Wait for Bind Result状態へ遷移する。オペレーションは他のSDF(消費側SDF)に送信される。SDSM-ShSSiは消費側SDFからの応答を待つ。DSAShadowBindオペレーションが成功した場合は、以前に消費側SDFに対して発行されたDSAShadowBindに対する応答の受信((E14)SDF_Bind_Success)により、SDFの状態はSDF Boundという状態へ遷移する。その他の場合は、SDSM-ShSSiはエラーの受信((E13)SDF_Bind_Error)によりIdle状態へ戻る。

17.49.2.1.1.2 CoordinateShadowUpdate とともに送信した DSAShadowBind

S D F 前条件 :

- (1) SDSM-ShSSi : Idle

S D F 後条件 :

- (1) SDSM-ShSSi : Wait for Coordination Result (DSAShadowBind が成功した場合)
- (2) SDSM-ShSSi : Idle (DSAShadowBind が失敗した場合)

SDSM-ShSSi が Idle 状態にあり、コピーの供給が必要である場合、内部イベントが生じる。このイベントは(e1) Bind to_Consumer と呼ばれ、Wait for Subsequent Requests 状態への遷移を引き起こし、他のオペレーションは待たされる。DSAShadowBind オペレーションと同時に送信すべき後続のオペレーションを許容するため、DSAShadowBind の送信は遅延する。Wait for Subsequent Requests 状態での CoordinateShadowUpdate オペレーションの受信は、(e2)Shadow_Coordinate_to Consumer という内部イベントを経て、Bind with Coordinate Shadow 状態への遷移を引き起こす。

Bind with Coordinate Shadow 状態でデリミタが受信されると、(e5)Send_Bind_with_CoordShadow という内部イベントを経て、Bind with CoordinateShadow Only 状態へ遷移する。これにより、二つのオペレーションは同時に他の S D F (消費側 S D F) に送信される。SDSM-ShSSi は消費側 S D F からの応答を待つ。DSAShadowBind オペレーションが成功した場合は、以前に消費側 S D F に対して発行された DSAShadowBind オペレーションに対する応答 (E7)SDF_Bind_Success の受信により、Wait for Coordination Result という状態へ遷移する。その他の場合は、SDSM-ShSSi はエラーの受信 (E6)SDF_Bind_Error) により Idle 状態へ戻る。

17.49.2.1.1.3 CoordinateShadowUpdate と UpdateShadow とともに送信した DSAShadowBind

S D F 前条件 :

- (1) SDSM-ShSSi : Idle

S D F 後条件 :

- (1) SDSM-ShSSi : Bound with CoordinateShadow Sent (DSAShadowBind が成功した場合)
- (2) SDSM-ShSSi : Idle (DSAShadowBind が失敗した場合)

SDSM-ShSSi が Idle 状態にあり、コピーの供給が必要である場合、内部イベントが生じる。このイベントは(e1)Bind_to_Consumer と呼ばれ、Wait for Subsequent Requests 状態への遷移を引き起こし、他のオペレーションは待たされる。DSAShadowBind オペレーションと同時に送信すべき後続のオペレーションを許容するため、DSAShadowBind の送信は遅延する。Wait for Subsequent Requests 状態での CoordinateShadowUpdate オペレーションの受信は、(e2)Shadow_Coordinate_to Consumer という内部イベントを経て、Bind with Coordinate Shadow 状態への遷移を引き起こす。

後続の UpdateShadow オペレーションの受信は、(e4) Update_to_Consumer といった内部イベントを通して、Bind with CoordinateShadow and Update 状態への遷移を引き起こす。これにより、三つのオペレーションは同時に他の S D F (消費側 S D F) に送信される。SDSM-ShSSi は消費側 S D F からの応答を待つ。DSAShadowBind オペレーションが成功した場合は、以前に消費側 S D F に発行した DSAShadowBind オペレーションに対する応答である(E9) SDF_Bind_Success の受信により、Bound with Coordinate Shadow Sent という状態へ遷移する。その他の場合は、SDSM-ShSSi はエラーの受信 (E8)SDF_Bind_Error) により Idle 状態へ戻る。

17.49.2.1.2 消費側起動による DSAShadowBind

S D F 前条件 :

- (1) SDSM-ShSCi : Idle

S D F 後条件 :

- (1) SDSM-ShSCi : SDF Bound (DSAShadowBind が成功した場合)
- (2) SDSM-ShSCi : Idle (DSAShadowBind が失敗した場合)

SDF は最初 Idle 状態にある。SDF (消費側 SDF) からの DSAShadowBind オペレーションの受信により引き起こされる(E1) Bind_from_Consumer という外部イベントの受信後、Wait for Bind Result 状態へ遷移する。DSAShadowBind オペレーションは、RequestShadowUpdate と同時に受信してもよい。この場合は、(E3)Request_from_Consumer といった外部イベントを経て同じ状態に留まる。SDF は DSAShadowBind アーギュメントの内容に従って DSAShadowBind オペレーションを実行する。SDF が DSAShadowBind オペレーションを完了させると、結果応答あるいはエラーが消費側 SDF に対して返される。SDF は DSAShadowBind が失敗した場合には Idle 状態に戻り、成功した場合には SDF Bound 状態へ遷移する。

17.49.2.2 エラー処理

シャドウイングオペレーションに関連するエラーの一般的な扱いは、ITU - T 勧告 X.525 の 12 章に、オペレーションエラーを通知するための TC サービスは、18.1 節に記述されている。

17.49.3 消費側エンティティ (SDF)

17.49.3.1 正常手順

17.49.3.1.1 供給側起動による DSAShadowBind

SDF 前条件 :

- (1) SDSM-ShCSi : Idle

SDF 後条件 :

- (1) SDSM-ShCSi : SDF Bound(DSAShadowBind が成功した場合)
- (2) SDSM-ShCSi : Idle(DSAShadowBind が失敗した場合)

SDF は最初 Idle 状態にある。SDF (供給側 SDF) からの DSAShadowBind オペレーションの受信により引き起こされる(E1)Bind from Supplier 結合という外部イベントの受信後、Wait for Bind Result 状態へ遷移する。DSAShadowBind オペレーションは、CoordinateShadowUpdate オペレーションまたは CoordinateShadowUpdate オペレーションおよび UpdateShadow オペレーションと同時に受信することができる。このような場合、(E3)Request from Supplier という外部イベントを経て同じ状態への遷移が 1 度または 2 度生じる。SDF は DSAShadowBind アーギュメントの内容に従って DSAShadowBind オペレーションを実行する。SDF が DSAShadowBind オペレーションを完了させると、結果応答あるいはエラーが供給側 SDF に対して返される。SDF は DSAShadowBind が失敗した場合には Idle 状態に戻り、成功した場合には SDF Bound 状態へ遷移する。

17.49.3.1.2 消費側起動による DSAShadowBind

17.49.3.1.2.1 自身の送信した DSAShadowBind

SDF 前条件 :

- (1) SDSM-ShCCi : Idle

SDF 後条件 :

- (1) SDSM-ShCCi : SDF Bound (DSAShadowBind が成功した場合)
- (2) SDSM-ShCCi : Idle (DSAShadowBind が失敗した場合)

SDSM-ShCCi が Idle 状態にあり、更新の要求が必要である場合、内部イベントが生じる。このイベントは(e1)Bind to Supplier と呼ばれ、Wait for Subsequent Requests 状態への遷移を引き起こし、他のオペレーションは待たされる。これら後続オペレーションが DSAShadowBind と同時に送信できるように、DSAShadowBind の送信は遅延される。アプリケーションプロセスがデリミタによって DSAShadowBind の送信を指示しない限り、SDSM-ShCCi は Wait

for Subsequent Requests 状態に留まり、オペレーションは送信されない。デリミタが受信されると(e3)Send Bind という内部イベントを経て Wait for Bind Result 状態へ遷移する。その時、オペレーションは他の S D F (供給側 S D F) に送信される。SDSM-ShCCi は供給側 S D F からの応答を待つ。DSAShadowBind オペレーションが成功した場合は、以前に供給側 S D F に対して発行された DSAShadowBind に対する応答の受信 (E7) SDF Bind Success) により、S D F の状態は SDF Bound という状態に遷移する。その他の場合は、SDSM-ShCCi はエラーの受信 (E6) SDF Bind Error) により Idle 状態へ戻る。

17.49.3.1.2.2 RequestShadowUpdate とともに送信した DSAShadowBind

S D F 前条件 :

- (1) SDSM-ShCCi : Idle

S D F 後条件 :

- (1) SDSM-ShCCi : Wait for RequestShadow Result (DSAShadowBind が成功した場合)
- (2) SDSM-ShCCi : Idle (DSAShadowBind が失敗した場合)

SDSM-ShCCi が Idle 状態にあり、更新の要求が必要である場合、内部イベントが生じる。このイベントは(e1)Bind to Supplier と呼ばれ、Wait for Subsequent Requests 状態への遷移を引き起こし、他のオペレーションは待たされる。これら後続のオペレーションを DSAShadowBind と同時に送信できるように、DSAShadowBind の送信は遅延される。(e2)Request to Supplier という内部イベントを経て Wait for Subsequent Requests 状態で RequestShadowUpdate を受信すると、Bind with RequestShadow 状態に遷移する。その時、二つのオペレーションは他の S D F (供給側 S D F) に送信される。SDSM-ShCCi は供給側 S D F からの応答を待つ。DSAShadowBind オペレーションが成功した場合は、以前に供給側 S D F に対して発行された DSAShadowBind オペレーションに対する応答の受信 (E5) SDF Bind Success) により、Wait for RequestShadow Result という状態に遷移する。その他の場合は、SDSM-ShCCi はエラーの受信 (E4) SDF Bind Error) により Idle 状態へ戻る。

17.49.3.2 エラー処理

シャドウイングオペレーションに関連するエラーの一般的な扱いは I T U - T 勧告 X . 5 2 5 の 12 章に、オペレーティングエラーを通知するための T C サービスは 18.1 節に記述されている。

17.50 in-DSAShadowUnbind 手順

17.50.1 一般記述

T T C 標準 J T - X 5 0 0 のシャドウイングオペレーションにより、2 つの S D F 間で情報がコピーされる。シャドウイングオペレーションはこのコピーされた情報を維持するためにも用いられる。2 つの S D F 間の個々のシャドウイング合意において、一方の S D F はコピーされた情報の供給側となり、他方の S D F は消費側となる。DSAShadowBind オペレーションと DSAShadowUnbind オペレーションは、コピーを供給するある期間のはじめと終わりに、協調して動作する S D F によって使用される。coordinateShadowUpdate オペレーションは、シャドウ供給側が更新を意図する対象のシャドウイング合意を示すために使用される。requestShadowUpdate オペレーションは、シャドウ消費側がシャドウ供給側からの更新を要求するために使用される。updateShadow オペレーションは、シャドウ供給側がコピーされたデータをシャドウ消費側に送信するために起動される。このオペレーションより前に、coordinateShadowUpdate あるいは requestShadowUpdate のオペレーションが行われなければならない。シャドウイングオペレーションの完全な記述については I T U - T 勧告 X . 5 2 5 を参照。

17.50.1.1 パラメータ

なし。

17.50.2 供給側エンティティ (S D F)

17.50.2.1 正常手順

17.50.2.1.1 供給側起動による DSAShadowUnBind

S D F 前条件 :

- (1) SDSM-ShSSi : SDF Bound
- (2) SDSM-ShSSi : Bound with CoordinateShadow Sent
- (3) SDSM-ShSSi : Wait for Coordination Result
- (4) SDSM-ShSSi : Wait for Update
- (5) SDSM-ShSSi : Wait for Update Confirmation

S D F 後条件 :

- (1) SDSM-ShSSi : Idle

SDSM-ShSSi は消費側 S D F に対して既に DSAShadowBind オペレーションを開始しており、SDF Bound、Bound with CoordinateShadow Sent、Wait for Coordination Result、Wait for Update あるいは Wait for Update Confirmation のいずれかの状態にある。S D F は S D F 間に確立された認証されたアソシエーションが終了すべきことを決定すると (例、ユーザの解放手順)、DSAShadowUnbind オペレーション ((e12)、(e15)、(e17)、(e20)あるいは(e22)SDF Unbind) を発行し、SDSM-ShSSi は Idle 状態に遷移する。

17.50.2.1.2 消費側起動による DSAShadowUnbind

S D F 前条件 :

- (1) SDSM-ShSCi : SDF Bound
- (2) SDSM-ShSCi : Wait for RequestShadow Result
- (3) SDSM-ShSCi : Wait for Update
- (4) SDSM-ShSCi : Wait for Update Confirmation

S D F 後条件 :

- (1) SDSM-ShSCi : Idle

DSAShadowBind オペレーションが既に発行され、SDSM-ShSCi は SDF Bound、Wait for RequestShadow Result、Wait for Update あるいは Wait for Update Confirmation のいずれかの状態で消費側 S D F からの要求 / 応答を待っているか、またはオペレーションを実行している。DSAShadowUnBind オペレーションを受信すると、SDF Unbind ((E6)、(E8)、(E12)あるいは(E15)) により、Idle 状態に遷移する。

17.50.2.2 エラー処理

シャドウイングオペレーションに関連するエラーの一般的な扱いは I T U - T 勧告 X . 5 2 5 の 12 章に、オペレーティングエラーを通知するための T C サービスは 18.1 節に記述されている。

17.50.3 消費側エンティティ (S D F)

17.50.3.1 正常手順

17.50.3.1.1 供給側起動による DSAShadowUnbind

S D F 前条件 :

- (1) SDSM-ShCSi : SDF Bound
- (2) SDSM-ShCSi : Wait for Coordination Result
- (3) SDSM-ShCSi : Wait for Update

(4) SDSM-ShCSi : Wait for Update Confirmation

S D F 後条件 :

(1) SDSM-ShCSi : Idle

DSAShadowBind オペレーションが既に発行され、SDSM-ShCSi は SDF Bound、Wait for Coordination Result、Wait for Update あるいは Wait for Update Confirmation のいずれかの状態で供給側 S D F からの要求を待っているか、またはオペレーションを実行している。DSAShadowUnbind オペレーションを受信すると、SDF Unbind ((E6)、(E9)、(E12)あるいは(E14)) により Idle 状態に遷移する。

17.50.3.1.2 消費側起動による DSAShadowUnbind

S D F 前条件 :

- (1) SDSM-ShCCi : SDF Bound
- (2) SDSM-ShCCi : Wait for RequestShadow Result
- (3) SDSM-ShCCi : Wait for Update
- (4) SDSM-ShCCi : Wait for Update Confirmation

S D F 後条件 :

(1) SDSM-ShCCi : Idle

SDSM-ShCCi は供給側 S D F に対して既に DSAShadowBind オペレーションを開始しており、SDF Bound、Wait for RequestShadow Result、Wait for Update あるいは Wait for Update Confirmation のいずれかの状態にある。S D F は S D F 間に確立された認証されたアソシエーションが終了すべきことを決定すると (例、ユーザの解放手順)、DSAShadowUnbind オペレーション ((e8)、(e10)、(e13)あるいは(e15)SDF Unbind) を発行し、SDSM-ShCCi は Idle 状態に遷移する。

17.51	EntityReleased 手順	#
17.52	EstablishChargingRecord 手順	#
17.53	EstablishTemporaryConnection 手順	#
17.54	EventNotificationCharging 手順	#
17.55	EventReportBCSM 手順	#
17.56	EventReportFacility 手順	#

17.57 Execute 手順

17.57.1 一般記述

Execute オペレーションは、入力情報を使い、既定の方法に従って一連の実行ステップを行い、結果情報を戻す。各ステップは D A P オペレーション (execute オペレーションであるかもしれない) か、アルゴリズムあるいはデシジョン試験の実行のいずれかである。

Execute オペレーションは D I T に常駐する S D F のあるエントリ上のメソッドを実行し、そのメソッドの結果から決定される選択情報を戻すために使用される。メソッドは一連の複合 S D F オペレーションを単一の S C F - S D F 間オペレーションに要約するために使用される。これは S D F のエントリ・メソッドに対してデータアクセススクリプトを関連付けて行う。このスクリプトは、後続のオペレーションを行うために必要な追加論理と共に、S C F - S D F インタフェース上で呼び出すことのできる一つ以上のオペレーションを扱うことができる。この追加論理には意思決定 (decision making)、データ操作 (data manipulation) を含む事が出来る。このスクリプトは入力値パラメータの一部を、スクリプトが呼び出すある内部オペレーションのパラメータとして使用

することができる。

S C Fはサービスに適したロジックを提供し、S S Fの呼制御機能を起動させる。

#

17.57.1.1 パラメータ

- object :
このパラメータは、メソッドが実行される、ディレクトリ情報ツリー(D I T)にあるS D Fのエントリを識別する。メソッドはこのエントリから / 上で実行される。このパラメータの使用方法についてはI T U - T 勧告 X . 5 0 1 の 9 章に定義されている。
- method-id :
このパラメータは、エントリ上で実行されるメソッドを識別する。このパラメータの値は、データスキーマで定義されているメソッドのA S N . 1 の記述から識別され、関連オブジェクトに固有である。
- inputAssertions :
このパラメータは、メソッド実行への入力として使用される属性値のセットを提供する。
- input-Attributes :
このフィールドは、メソッド実行への入力として依頼されてもよい属性を識別する。
- specificInput :
このパラメータは、メソッドを実行するためにエントリ上で必要とされる追加情報を識別する。このパラメータの種別は、データスキーマで定義されているメソッドのA S N . 1 の記述により識別される。
- outputAssertions :
このパラメータは、メソッド実行の結果として戻される属性値を含む。
- output-Attributes :
このフィールドは、メソッド実行への出力として戻されてもよい属性を識別する。
- specificOutput :
このパラメータは、エントリ上のメソッドを呼び出す際に実行したオペレーションの結果として戻される情報を含む。このパラメータの種別は、データスキーマで定義されているメソッドのA S N . 1 の記述により識別される。
- CommonArguments :
T T C 標準 J T - X 5 0 0 のすべてのオペレーションはセキュリティ情報などの共通のアーギュメントを持っている。これらのアーギュメントの使用方法についてはI T U - T 勧告 X . 5 1 1 の 7.3 節に定義されている。

17.57.2 起動側エンティティ (S C F)

17.57.2.1 正常手順

S C F 前条件 :

- (1) SCSM : SDF Bound あるいは、Wait for Subsequent Requests

S C F 後条件 :

- (1) SCSM : SDF Bound

SCSM が Wait for Subsequent Requests 状態にあり、サービス論理が S D F でエントリメソッドを実行する必要がある場合、内部イベント ((e2)Request to SDF) が生じる。アプリケーションプロセスがデリミタによってオペレーションの送信を指示しない限り、SCSM は Wait for Subsequent Requests 状態に留まり、オペレーションは送信されない。デリミタが受信されると、オペレーションは Bind アーギュメントを含んだ一つのメッセージとして S D F に送信される。SCSM は S D F からの応答を待つ。以前に S D F に対して発行された Bind オペレーションに対する応答の受信 ((E5)Response from SDF with Bind または (E4)Bind Error) により、S C F の状態は SDF Bound 状態または Idle 状態に遷移する。SCSM が Idle 状態に遷移すると、Execute オペレーションは破棄される。SDF Bound 状態で、Execute オペレーションの応答 ((E7)Response from SDF) により、S C F は同じ状態 (SDF Bound) に遷移する。S D F からの応答は Execute オペレーションの結果、あるいはエラーであるかもしれない。

SCSM が SDF Bound 状態で、サービス論理が S D F 上でエントリメソッドを実行する必要がある場合、内部イベント ((e6)Request to SDF) が生じる。このイベントにより、同じ SDF Bound 状態に遷移し、SCSM は S D F からの応答を待つ。以前に S D F に対して発行された Execute オペレーションに対する応答の受信 ((E7)Response from SDF) により、S C F は SDF Bound という同じ状態に遷移する。S D F からの応答は Execute オペレーションの結果、あるいはエラーであるかもしれない。

17.57.2.2 エラー処理

Execute オペレーションに関連するエラーの一般的な扱いは I T U - T 勧告 X . 5 1 1 の 12 章に、オペレーティングエラーを通知するための T C サービスは 18 章に記述されている。

17.57.3 応答側エンティティ (S D F)

17.57.3.1 正常手順

S D F 前条件 :

- (1) SDSM : SCF Bound あるいは、Bind Pending

S D F 後条件 :

- (1) SDSM : SCF Bound

S D F は Bind Pending 状態にあり、S C F からの Execute オペレーションの受信により引き起こされる (E3)Request from SCF という外部イベントが生じる。S D F は Bind オペレーションが成功するまでオペレーションを行わずに同じ状態に留まる。

S D F は SCF Bound 状態にあり、S C F からの Execute オペレーションの受信により引き起こされる (E7)Request from SCF という外部イベントが生じる。S D F はオペレーションが実行されるのを待つ。

(E7)というイベントを受信すると、オペレーションパラメータで指定されたデータを検索する前に、S D F は次の動作を取る。

- S D F は要求によりアクセスされたオブジェクトが存在するか確認する。
- S D F はオペレーションで参照されるメソッドがオブジェクトに存在し、アーギュメントの型が正しいかを確認する。

- S D F は要求を実行してもらったユーザがエン트리上のメソッドを実行するのに十分なアクセス権を有するか確認する。

これらの動作が完了すると、S D F は S D F のエン트리メソッドに関連するデータアクセススクリプトを実行する。スクリプトにより起動されるそれぞれの内部オペレーションを実行する前に、S D F は次の動作を取る。

- S D F は要求を実行してもらったユーザがそのオペレーションを実行するのに十分なアクセス権を有するか確認する。
- S D F はオペレーションを実行すべきオブジェクト、属性、メソッドが D I T に存在するか確認する。

上記すべての動作が完了すると、S D F はスクリプトの結果を S C F に戻す。この結果の送信が(e6)Response to SCF というイベントにあたる。

17.57.3.2 エラー処理

Execute オペレーションに関連するエラーの一般的な扱いは I T U - T 勧告 X . 5 1 1 の 12 章に記載されている。

メソッドを実行しているエントリに基本アクセス制御 (basic-access-control) が作動している場合は、次のアクセス制御のシーケンスを適用する。

- ExecuteMethod の許可が、メソッドを実行しているエントリに必要なものである。許可が与えられていない場合、I T U - T 勧告 X . 5 1 1 の 7.11.3 節に従ってオペレーションは失敗する。この場合、insufficient AccessRights あるいは noInformation を設定した problem による SecurityError を戻す。
- エン트리メソッドに関連したデータアクセススクリプトが起動するそれぞれの内部オペレーションが実行される前に、オペレーションがあたかも S C F - S D F インタフェース上で呼び出されたかのように基本アクセス制御が適用される。これらの内部オペレーションのいずれかが失敗した場合、全てのオペレーションは失敗し、executionFailure を設定した problem による executionError を戻す。

オペレーションがその他のデータアクセス論理により失敗した場合は、executionFailure を設定した problem による executionError を戻す。

オペレーションが入力値の型または値が不正なために失敗した場合は、missinginputValues を設定した problem による executionError を戻す。

存在しないエントリ上でメソッドを実行するように要求された場合、noSuchObject を設定した problem による NameError を戻す。

オペレーティングエラーを通知するための T C サービスは 18 章に記載されている。

17.58	FacilitySelectedAndAvailable 手順	#
17.59	FurnishChargingInformation 手順	#
17.60	HandlingInformationRequest 手順	#

17.61 HandlingInformationResult 手順 #

17.62 HoldCallInNetwork 手順 #

17.63 in-DSAUnbind 手順

17.63.1 一般記述

in-DSAUnbind オペレーションは起動側 S D F により使用され、エンドユーザに代わって、起動側 S D F と応答側 S D F 間の認証された連鎖アソシエーションを終了させる。

17.63.1.1 パラメータ

なし。

17.63.2 起動側エンティティ(S D F)

17.63.2.1 正常手順

S D F 前条件 :

- (1) 起動側 S D F が、オペレーションの連鎖を要求しているエンドユーザアソシエーションが解放されたことを指示する Unbind オペレーションを受信した。
- (2) SDSM-ChI : SDF Bound

S D F 後条件 :

- (1) SDSM-ChI : Idle

SDSM-ChI は応答側 S D F に対してすでに Bind オペレーションを開始しており、SDF Bound 状態にある。起動側 S D F は応答側 S D F に対して認証された連鎖アクセスが終了すべき指示を受信している。SDSM-ChI は in-DSAUnbind オペレーション ((e6) DSAUnbind_to_SDF) を発行し、Idle 状態に遷移する。

17.63.2.2 エラー処理

Unbind オペレーションはエラーに関連するオペレーションを持たない。

17.63.3 応答側エンティティ(S D F)

17.63.3.1 正常手順

S D F 前条件 :

- (1) SDSM-ChT : SDF Bound

S D F 後条件 :

- (1) SDSM-ChT : Idle

dSABind が以前に発行され、SDSM-ChT は、SDF Bound 状態で起動側 S D F からの要求を待っているか、またはオペレーションを実行している。in-DSAUnbind オペレーションを受信すると(E5)DSAUnBind_from_SDF により Idle 状態に遷移する。

17.63.3.2 エラー処理

Unbind オペレーションはエラーに関連するオペレーションを持たない。

17.64 InitialDP 手順 #

17.65 InitiateAssociation 手順 #

17.66	InitiateCallAttempt 手順	#
17.67	ManageTriggerData 手順	#
17.68	MergeCallSegments 手順	#

17.69 ModifyEntry 手順

17.69.1 一般記述

TTC 標準 JT - X 5 0 0 の ModifyEntry オペレーションはデータオブジェクトに対して、一つあるいは複数の変更を施すことを SDF に要求するために使用される。変更はオブジェクトを構成する属性とその値に対して行われる。実行される変更種別は、SCF が指定するオペレーションアークギュメント中で与えられる。変更は、オブジェクトを識別するために使用される属性(即ちオブジェクト名)に対して行われることはない。ModifyEntry オペレーションの完全な記述については ITU - T 勧告 X . 5 1 1 の 11.3 節を参照。

17.69.1.1 パラメータ

ITU - T 勧告 X . 5 1 1 の 11.3.2 及び 11.3.3 節を参照。

17.69.2 起動側エンティティ(SCF)

17.69.2.1 正常手順

SCF 前条件：

- (1) SCSCM： SDF Bound あるいは、Wait for Subsequent Requests

SCF 後条件：

- (1) SCSCM： SDF Bound

SCSCM が Wait for Subsequent Requests 状態にあり、SDF に管理されている情報の更新がサービス論理において必要である場合、内部イベント (e2) Request_to_SDF が生じる。アプリケーションプロセスが、デリミタ(またはタイマの満了)によってオペレーションの送信を指示しない限り、SCSCM は Wait for Subsequent Requests 状態に留まりオペレーションは送信されない。オペレーションは Bind アークギュメントを含んだメッセージによって SDF に送信される。SCSCM は SDF からの応答を待つ。以前に SDF に対して発行された Bind オペレーションに対する応答の受信((E5) Response_from_SDF_with_Bind あるいは(E4) Bind_error)を受けて、SCF は SDF Bound または Idle 状態へ遷移する。SCSCM が Idle 状態に遷移した場合、ModifyEntry オペレーションは破棄される。SDF Bound 状態において ModifyEntry オペレーションの応答 (E7) Response_from_SDF を受けると SCF は同一の状態 (SDF Bound) へ遷移する。その応答は、ModifyEntry オペレーションの結果、あるいはエラーであるかもしれない。

SCSCM が SDF Bound 状態にあり、SDF に管理されている情報の更新がサービス論理において必要である場合、内部イベントが生じる。このイベントは(e6) Request_to_SDF と呼ばれ、全く同じ SDF Bound 状態への遷移を引き起こし、SCSCM は SDF からの応答を待つ。以前に SDF に対して発行された ModifyEntry オペレーションに対する応答の受信((E7) Response_from_SDF)を受けて、SCF は全く同じ SDF Bound 状態へ遷移する。SDF からの応答は、ModifyEntry オペレーションの結果、あるいはエラーであるかもしれない。

17.69.2.2 エラー処理

オペレーションに関連するエラーの一般的な扱いは ITU - T 勧告 X . 5 1 1 の 11.3.4 節及び 11.3.5 節に、オペレーションエラーを通知するための TC サービスは 18 章に記述されている。

17.69.3 応答側エンティティ(S D F)

17.69.3.1 正常手順

S D F 前条件 :

- (1) SDSM : SCF Bound あるいは、Bind Pending

S D F 後条件 :

- (1) SDSM : SCF Bound

S D F は Bind Pending 状態にあり、S C F からの ModifyEntry オペレーションの受信により引き起こされる(E3) Request_from_SCF という外部イベントが生ずる。S D F は Bind オペレーションの実行が成功するまでオペレーションを処理せず、同一状態に留まる。

S D F は SCF Bound 状態にあり、S C F からの ModifyEntry オペレーションの受信により引き起こされる(E7) Request_from_SCF という外部イベントが生ずる。S D F はオペレーションが実行されるのを待つ。

イベント(E7)を受信すると、オペレーションのパラメータに規定される属性の更新を行う前に S D F は以下の動作を行う。

- 要求に関するオブジェクトが存在することの検証
- 要求はユーザに代わって実行されるが、ユーザがオペレーションに含まれる個々の変更をオブジェクトに対して行うための十分なアクセス権を有しているかどうかの検証
- オブジェクト中にオペレーションが実行されるべき属性あるいは値が存在するかどうかの検証
- 要求された変更がオブジェクトの属するオブジェクトクラスあるいは属性の構文と矛盾しないこととの検証

上述の動作がオペレーションで要求される全ての変更に対して完全に行われた後、同一の属性に対する全ての変更が changes パラメータに与えられた順序に従って行われる。結果応答が S C F に返される。結果応答の送信は(e6) Response_to_SCF というイベントに対応する。

17.69.3.2 エラー処理

オペレーションに関連するエラーの一般的な扱いは I T U - T 勧告 X . 5 1 1 の 11.3.4 節及び 11.3.5 節に、オペレーションエラーを通知するための T C サービスは 18 章に記述されている。

17.70	MoveCallSegments 手順	#
17.71	MoveLeg 手順	#
17.72	NetworkCapability 手順	#
17.73	NotificationProvided 手順	#
17.74	OAbandon 手順	#
17.75	OAnswer 手順	#
17.76	OCalledPartyBusy 手順	#
17.77	ODisconnect 手順	#
17.78	OMidCall 手順	#
17.79	ONoAnswer 手順	#
17.80	OriginationAttempt 手順	#
17.81	OriginationAttemptAuthorized 手順	#
17.82	OSuspended 手順	#
17.83	PlayAnnouncement 手順	#

17.84	PromptAndCollectUserInformation 手順	#
17.85	PromptAndReceiveMessage 手順	#
17.86	ProvideUserInformation 手順	#
17.87	Reconnect 手順	#
17.88	ReleaseAssociation 手順	#
17.89	ReleaseCall 手順	#

17.90 RemoveEntry 手順

17.90.1 一般記述

TTC 標準 JT - X 5 0 0 の RemoveEntry オペレーションは DIT からの葉エントリ (オブジェクトエントリあるいはエリアエントリ) の削除を SDF に要求するために用いられる。RemoveEntry オペレーションの完全な記述については ITU - T 勧告 X . 5 1 1 の 11.2 節を参照。

17.90.1.1 パラメータ

ITU - T 勧告 X . 5 1 1 の 11.2.2 及び 11.2.3 節を参照。

17.90.2 起動側エンティティ (SCF)

17.90.2.1 正常手順

SCF 前条件 :

- (1) SCSCM : SDF Bound あるいは、Wait for Subsequent Requests

SCF 後条件 :

- (1) SCSCM : SDF Bound

SCSCM が Wait for Subsequent Requests 状態にあり、SDF からのエントリの削除がサービス論理において必要である場合、内部イベント ((e2) Request_to_SDF) が生じる。アプリケーションプロセスが、デリミタ (またはタイマの満了) によってオペレーションの送信を指示しない限り、SCSCM は Wait for Subsequent Requests 状態に留まりオペレーションは送信されない。オペレーションは Bind アーギュメントを含んだメッセージによって SDF に送信される。SCSCM は SDF からの応答を待つ。以前に SDF に対して発行された Bind オペレーションに対する応答の受信 ((E5) Response_from_SDF_with_Bind あるいは (E4) Bind_error) を受けて、SCF は SDF Bound または Idle 状態へ遷移する。SCSCM が Idle 状態に遷移した場合、RemoveEntry オペレーションは破棄される。SDF Bound 状態において RemoveEntry オペレーションの応答 ((E7) Response_from_SDF) を受けると SCF は同一の状態 (SDF Bound) へ遷移する。その応答は、RemoveEntry オペレーションの結果、あるいはエラーであるかもしれない。

SCSCM が SDF Bound 状態にあり、SDF からのエントリの削除がサービス論理において必要である場合、内部イベントが生じる。このイベントは (e6) Request_to_SDF と呼ばれ、全く同じ SDF Bound 状態への遷移を引き起こし、SCSCM は SDF からの応答を待つ。以前に SDF に対して発行された RemoveEntry オペレーションに対する応答の受信 ((E7) Response_from_SDF) を受けて、SCF は全く同じ SDF Bound 状態へ遷移する。SDF からの応答は、RemoveEntry オペレーションの結果、あるいはエラーであるかもしれない。

17.90.2.2 エラー処理

オペレーションに関連するエラーの一般的な扱いは ITU - T 勧告 X . 5 1 1 の 11.2.4 節及び 11.2.5 節に、オペレーションエラーを通知するための TC サービスは 18 章に記述されている。

17.90.3 応答側エンティティ(S D F)

17.90.3.1 正常手順

S D F 前条件 :

- (1) SDSM : SCF Bound あるいは、Bind Pending

S D F 後条件 :

- (1) SDSM : SCF Bound

S D F は Bind Pending 状態にあり、S C F からの RemoveEntry オペレーションの受信により引き起こされる(E3) Request_from_SCF という外部イベントが生ずる。S D F は Bind オペレーションの実行が成功するまでオペレーションを処理せず、同一状態に留まる。

S D F が SCF Bound 状態にあり、S C F からの RemoveEntry オペレーションの受信により引き起こされる(E7) Request_from_SCF という外部イベントが生ずる。S D F はオペレーションが実行されるのを待つ。

イベント(E7)を受信すると、エントリを削除する前に、S D F は以下の動作を行う。

- 削除されるエントリが存在しかつ葉エントリであることの検証
- エントリを削除するためのアクセス権が十分であることの検証

上述の動作が完全に行われた後、S D F のデータベースからエントリが削除される。S C F に対して空の結果が返送される。結果応答の送信は(e6) Response_to_SCF というイベントに対応する。

17.90.3.2 エラー処理

オペレーションに関連するエラーの一般的な扱いは I T U - T 勧告 X . 5 1 1 の 11.2.4 節及び 11.2.5 節に、オペレーションエラーを通知するための T C サービスは 18 章に記述されている。

17.91	ReportChargingInformation 手順	#
17.92	ReportUTSI 手順	#
17.93	RequestCurrentStatusReport 手順	#
17.94	RequestEveryStatusChangeReport 手順	#
17.95	RequestFirstStatusMatchReport 手順	#
17.96	RequestNotification 手順	#
17.97	RequestNotificationChargingEvent 手順	#
17.98	RequestReportBCSMEEvent 手順	#
17.99	RequestReportBCUSMEEvent 手順	#
17.100	RequestReportFacilityEvent 手順	#
17.101	RequestReportUTSI 手順	#

17.102 RequestShadowUpdate 手順

17.102.1 一般記述

T T C 標準 J T - X 5 0 0 のシャドウイングオペレーションにより、2 つの S D F 間で情報がコピーされる。シャドウイングオペレーションはコピーされた情報を維持するためにも用いられる。2 つの S D F 間の個々のシャドウイング合意において、一方の S D F はコピーされた情報の供給側となり、他方の S D F は消費側となる。

DSAShadowBind オペレーションと DSAShadowUnbind オペレーションは、コピーを供給するある期間のはじめと終わりに、協調して動作する S D F によって使用される。coordinateShadowUpdate オペレーションは、シャドウ供給側が更新を意図する対象のシャドウイング合意を示すために使用される。requestShadowUpdate オペレーションは、シャドウ消費側がシャドウ供給側からの更新を要求するために使用される。updateShadow オペレーション

ンは、シャドウ供給側がコピーされるデータをシャドウ消費側に送信するために起動される。このオペレーションより前に、coordinateShadowUpdate あるいは requestShadowUpdate のオペレーションが行われなければならない。シャドウイングオペレーションの完全な記述については I T U - T 勧告 X . 5 2 5 を参照。

17.102.1.1 パラメータ

requestShadowUpdate オペレーションについては、I T U - T 勧告 X . 5 2 5 の 11.2 節を参照。

17.102.2 供給側エンティティ(S D F)

17.102.2.1 正常手順

17.102.2.1.1 自身の受信した RequestShadowUpdate

S D F 前条件：

- (1) SDSM-ShSCi : SDF Bound

S D F 後条件：

- (1) SDSM-ShSCi : Wait for Update (RequestShadowUpdate が成功した場合)
- (2) SDSM-ShSCi : SDF Bound (RequestShadowUpdate が失敗した場合)

S D F は最初 SDF Bound 状態にある。消費側 S D F からの requestShadowUpdate オペレーションの受信により引き起こされる(E7) Request_for_Shadow_from_Consumer という外部イベントの受信後、Wait for RequestShadow Result 状態へ遷移する。S D F は requestShadowUpdate アーギュメントの内容に従って requestShadowUpdate オペレーションを実行する。S D F が requestShadowUpdate オペレーションを完了させると、結果応答あるいはエラーが消費側 S D F に対して返される。S D F は requestShadowUpdate が失敗した場合には SDF Bound 状態に戻り、成功した場合には Wait for Update 状態へ遷移する。

17.102.2.1.2 CoordinateShadowUpdate と共に送信した DSAShadowBind

17.102.2.1.3 DSAShadowBind と共に受信した RequestShadowUpdate

S D F 前条件：

- (1) SDSM-ShSCi : Wait for Bind Result

S D F 後条件：

- (1) SDSM-ShSCi : Wait for Update (RequestShadowUpdate が成功した場合)
- (2) SDSM-ShSCi : SDF Bound (RequestShadowUpdate が失敗した場合)

S D F は最初、DSAShadowBind オペレーション以外に受信すべき他のオペレーションを待つ Wait for Bind Result 状態にある。RequestShadowUpdate オペレーション受信時、外部イベント(E3)Request_from_Consumer により、同じ状態へ遷移する。S D F は DSAShadowBind オペレーションを実行し、内部イベント(e5)SDF_Bind_Success により SDF Bound 状態へ遷移する。RequestShadowUpdate オペレーションを既に受信しているため、外部イベント(E7) Request_for_Shadow_from_Consumer により Wait for RequestShadow Result 状態へ遷移する。その後 S D F は requestShadowUpdate アーギュメントの内容に従って RequestShadowUpdate オペレーションを実行する。S D F が RequestShadowUpdate オペレーションを完了させると、結果応答あるいはエラーが消費側 S D F に対して返される。S D F は RequestShadowUpdate が失敗した場合には SDF Bound 状態に戻り、成功した場合には Wait for Update 状態へ遷移する。

17.102.3 消費側エンティティ(S D F)

17.102.3.1 正常手順

17.102.3.1.1 自身の送信した RequestShadowUpdate

S D F 前条件 :

- (1) SDSM-ShCCi : SDF Bound

S D F 後条件 :

- (1) SDSM-ShCCi : Wait for Update (RequestShadowUpdate が成功した場合)
- (2) SDSM-ShCCi : SDF Bound (RequestShadowUpdate が失敗した場合)

SDSM-ShCCi が SDF Bound 状態にあり、シャドウの更新の要求が必要である場合、内部イベントが生じる。このイベントは(e9)Shadow_Request_to_Supplier と呼ばれ、Wait for RequestShadow Result 状態への遷移を引き起こし、供給側 S D F へオペレーションが送信される。SDSM-ShCCi は供給側 S D F からの応答を待つ。RequestShadowUpdate オペレーションが成功した場合は、以前に供給側 S D F に対して発行された RequestShadowUpdate オペレーションに対する応答の受信 ((E12)Request_Shadow_Result) により、S D F は Wait for Update 状態へ遷移する。その他の場合は、SDSM-ShCCi はエラーの受信 ((E11)Failed_Shadow_Request) により SDF Bound 状態へ戻る。

17.102.3.1.2 DSAShadowBind と共に送信した RequestShadowUpdate

S D F 前条件 :

- (1) SDSM-ShCCi : Wait for Subsequent Requests

S D F 後条件 :

- (1) SDSM-ShCCi : Wait for Update (RequestShadowUpdate が成功した場合)
- (2) SDSM-ShCCi : SDF Bound (RequestShadowUpdate が失敗した場合)

SDSM-ShCCi が Wait for Subsequent Requests 状態にあり、シャドウの更新の要求が必要である場合、内部イベントが生じる。このイベントは(e2)Request_to_Supplier と呼ばれ、Bind with RequestShadow 状態への遷移を引き起こし、供給側 S D F へオペレーションが送信される。SDSM-ShCCi は供給側 S D F からの応答を待つ。DSAShadowBind 成功時、イベント(E5)SDF_Bind_Success により Wait for RequestShadow Result 状態へ遷移する。その後 SDSM-ShCCi は供給側からの応答を待つ。RequestShadowUpdate オペレーションが成功した場合は、以前に供給側 S D F に対して発行された RequestShadowUpdate に対する応答の受信 ((E12)Request_Shadow_Result) により、S D F は Wait for Update 状態へ遷移する。その他の場合は、SDSM-ShCCi はエラーの受信 ((E11)Failed_Shadow_Request) により SDF Bound 状態へ戻る。

17.102.3.2 エラー処理

シャドウイングオペレーションに関連するエラーの一般的な扱いは I T U - T 勧告 X . 5 2 5 の 12 章に、オペレーションエラーを通知するための T C サービスは 18.1 節に記述されている。

17.103	ResetTimer 手順	#
17.104	RouteSelectFailure 手順	#
17.105	SCFBind 手順	#
17.106	scfBind 手順 (連鎖の場合)	#
17.107	SCFUnBind 手順	#
17.108	scfUnBind 手順 (連鎖の場合)	#
17.109	ScriptClose 手順	#
17.110	ScriptEvent 手順	#
17.111	ScriptInformation 手順	#

17.113 Search 手順

17.113.1 一般記述

TTC 標準 JT - X 5 0 0 の Search オペレーションは、関心のあるエントリを探すために SDF 内の DIT の一部を検索し、それらのエントリから選択された情報を返送するために用いられる。Search オペレーションの完全な記述については ITU - T 勧告 X . 5 1 1 の 10.2 節を参照。

17.113.1.1 パラメータ

ITU - T 勧告 X . 5 1 1 の 10.2.2 及び 10.2.3 節を参照。

17.113.2 起動側エンティティ(SCF)

17.113.2.1 正常手順

SCF 前条件 :

- (1) SCSM : SDF Bound あるいは、Wait for Subsequent Requests

SCF 後条件 :

- (1) SCSM : SDF Bound

SCSM が Wait for Subsequent Requests 状態にあり、SDF に管理されている情報の検索あるいは参照がサービス論理において必要である場合、内部イベント ((e2)Request_to_SDF) が生じる。アプリケーションプロセスが、デリミタ(またはタイマの満了)によってオペレーションの送信を指示しない限り、SCSM は Wait for Subsequent Requests 状態に留まりオペレーションは送信されない。オペレーションは Bind アーギュメントを含んだメッセージによって SDF に送信される。SCSM は SDF からの応答を待つ。以前に SDF に対して発行された Bind オペレーションに対する応答の受信((E5) Response_from_SDF_with_Bind あるいは(E4) Bind_error)を受けて、SCF は SDF Bound または Idle 状態へ遷移する。SCSM が Idle 状態に遷移した場合、Search オペレーションは破棄される。SDF Bound 状態において Search オペレーションの応答 ((E7) Response_from_SDF) を受けると SCF は同一の状態 (SDF Bound) へ遷移する。その応答は、Search オペレーションの結果、あるいはエラーであるかもしれない。

SCSM が SDF Bound 状態にあり、SDF に管理されている情報の検索あるいは参照がサービス論理において必要である場合、内部イベントが生じる。このイベントは(e6)Request_to_SDF と呼ばれ、全く同じ SDF Bound 状態への遷移を引き起こし、SCSM は SDF からの応答を待つ。以前に SDF に対して発行された Search オペレーションに対する応答の受信((E7) Response_from_SDF)を受けて、SCF は全く同じ SDF Bound 状態へ遷移する。SDF からの応答は、Search オペレーションの結果、あるいはエラーであるかもしれない。

17.113.2.2 エラー処理

オペレーションに関連するエラーの一般的な扱いは ITU - T 勧告 X . 5 1 1 の 10.2.4 節及び 10.2.5 節に、オペレーションエラーを通知するための TC サービスは 18 章に記述されている。

17.113.3 応答側エンティティ(SDF)

17.113.3.1 正常手順

SDF 前条件 :

- (1) SDSM : SCF Bound あるいは、Bind Pending

SDF 後条件 :

(1) SDSM : SCF Bound

SDFは Bind Pending 状態にあり、SCFからの Search オペレーションの受信により引き起こされる(E3) Request_from_SCF という外部イベントが生ずる。SDFは Bind オペレーションの実行が成功するまでオペレーションを処理せず、同一状態に留まる。

SDFは SCF Bound 状態にあり、SCFからの Search オペレーションの受信により引き起こされる(E7) Request_from_SCF という外部イベントが生ずる。SDFはオペレーションが実行されるのを待つ。

イベント(E7)を受信すると、オペレーションのパラメータの規定に従ってデータの参照を行う前に、SDFは以下の動作を行う。

- 要求に関するオブジェクトが存在することの検証
- 要求はユーザに代わって実行されるが、ユーザがオペレーションの実行に際して迎られるオブジェクト及び属性にアクセスするための十分なアクセス権を有しているかどうかの検証
- オブジェクト中にオペレーションが実行されるべき属性が存在するかどうかの検証

上述の動作が完全に行われた後、SDFは参照規範を満足する全ての属性をSCFに返送する。結果応答の送信は(e6) Response_to_SCF というイベントに対応する。

17.113.3.2 エラー処理

オペレーションに関連するエラーの一般的な扱いはITU-T勧告X.511の10.2.4節及び10.2.5節に、オペレーションエラーを通知するためのTCサービスは18章に記述されている。

17.114	SelectFacility 手順	#
17.115	SelectRoute 手順	#
17.116	SendChargingInformation 手順	#
17.117	SendComponent 手順	#
17.118	SendFacilityInformation 手順	#
17.119	SendSTUI 手順	#
17.120	ServiceFilteringResponse 手順	#
17.121	SpecializedResourceReport 手順	#
17.122	SplitLeg 手順	#
17.123	StatusReport 手順	#
17.124	TAnswer 手順	#
17.125	TBusy 手順	#
17.126	TDisconnect 手順	#
17.127	TerminationAttempt 手順	#
17.128	TermAttemptAuthorized 手順	#
17.129	TMidCall 手順	#
17.130	TNoAnswer 手順	#
17.131	TSuspended 手順	#

17.132 UpdateShadow 手順

17.132.1 一般記述

TTTC標準JT-X500のシャドウイングオペレーションにより、2つのSDF間で情報がコピーされる。シャドウイングオペレーションはコピーされた情報を維持するためにも用いられる。2つのSDF間の個々のシ

シャドウイング合意において、一方の S D F はコピーされた情報の供給側となり、他方の S D F は消費側となる。

DSAShadowBind オペレーションと DSAShadowUnbind オペレーションは、コピーを供給するある期間のはじめと終わりに、協調して動作する S D F によって使用される。coordinateShadowUpdate オペレーションは、シャドウ供給側が更新を意図する対象のシャドウイング合意を示すために使用される。requestShadowUpdate オペレーションは、シャドウ消費側がシャドウ供給側からの更新を要求するために使用される。updateShadow オペレーションは、シャドウ供給側がコピーされるデータをシャドウ消費側に送信するために起動される。このオペレーションより前に、coordinateShadowUpdate あるいは requestShadowUpdate のオペレーションが行われなければならない。シャドウイングオペレーションの完全な記述については I T U - T 勧告 X . 5 2 5 を参照。

17.132.1.1 パラメータ

UpdateShadow オペレーションについては、I T U - T 勧告 X . 5 2 5 の 11.3 節を参照。

17.132.2 供給側エンティティ(S D F)

17.132.2.1 正常手順

17.132.2.1.1 供給側起動による UpdateShadow

17.132.2.1.1.1 自身の送信した UpdateShadow

S D F 前条件：

- (1) SDSM-ShSSi : Wait for Update

S D F 後条件：

- (1) SDSM-ShSSi : Wait for Update

SDSM-ShSSi が Wait for Update 状態にあり、シャドウの更新が必要である場合、内部イベントが生じる。このイベントは(e21)Shadow_Update_to_Consumer と呼ばれ、Wait for Update Confirmation 状態への遷移を引き起こし、消費側 S D F へオペレーションが送信される。SDSM-ShSSi は消費側 S D F からの応答を待つ。以前に消費側 S D F に対して発行された UpdateShadow に対する応答の受信 ((E23)Shadow_Update_Confirmed) により、S D F は Wait for Update 状態に遷移する。消費側 S D F からの応答は、UpdateShadow オペレーションの結果、あるいはエラーであるかもしれない。

17.132.2.1.1.2 DSAShadowBind と CoordinateShadowUpdate と共に送信した updateShadow

S D F 前条件：

- (1) SDSM-ShSSi : Bind with CoordinateShadow

S D F 後条件：

- (1) SDSM-ShSSi : Wait for Update

SDSM-ShSSi が Bind with CoordinateShadow 状態にあり、シャドウの更新が必要である場合、内部イベントが生じる。このイベントは(e4)Update_to_Consumer と呼ばれ、Bind with CoordinateShadow and Update 状態への遷移を引き起こし、DSAShadowBind 及び CoordinateShadowUpdate オペレーションにより消費側 S D F へオペレーションが送信される。その後 SDSM-ShSSi は消費側からの応答を待つ。以前発行された DSAShadowBind に対する応答の受信 ((E9)SDF_Bind_Success) により、Bound with Coordinate Shadow Sent 状態に遷移する。SDSM-ShSSi は、以前に消費側 S D F に対して発行された CoordinateShadowUpdate オペレーションに対する消費側からの応答を待つ。イベントの受信 ((E10)Shadow_Coordinate_Confirmed) により、Wait for Update Confirmation 状態に遷移する。以前に消費側 S D F に対して発行された UpdateShadow オペレーションに対する応答の受信 ((E23)Shadow_Update_Confirmed) により、S D F は Wait for Update 状態に遷移する。消費側 S D F からの応答は、UpdateShadow オペレーションの結果、あるいはエラーであるかもしれない。

17.132.2.1.2 消費側起動による UpdateShadow

17.132.2.1.2.1 自身の送信した UpdateShadow

S D F 前条件 :

- (1) SDSM-ShSCi : Wait for Update

S D F 後条件 :

- (1) SDSM-ShSCi : Wait for Update

SDSM-ShSCi が Wait for Update 状態にあり、シャドウの更新が必要である場合、内部イベントが生じる。このイベントは(e13)Shadow_Update_to_Consumer と呼ばれ、Wait for Update Confirmation 状態への遷移を引き起こし、消費側 S D F へオペレーションが送信される。SDSM-ShSCi は消費側 S D F からの応答を待つ。以前に消費側 S D F に対して発行された updateShadow に対する応答の受信 (E14)Shadow_Update_Confirmed) により、S D F は Wait for Update 状態に遷移する。消費側 S D F からの応答は、updateShadow オペレーションの結果、あるいはエラーであるかもしれない。

17.132.2.2 エラー処理

シャドウイングオペレーションに関連するエラーの一般的な扱いは I T U - T 勧告 X . 5 2 5 の 12 章に、オペレーションエラーを通知するための T C サービスは 18.1 節に記述されている。

17.132.3 消費側エンティティ(S D F)

17.132.3.1 正常手順

17.132.3.1.1 供給側起動による UpdateShadow

17.132.3.1.1.1 自身の受信した UpdateShadow

S D F 前条件 :

- (1) SDSM-ShCSi : Wait for Update

S D F 後条件 :

- (1) SDSM-ShCSi : Wait for Update

S D F は最初 Wait for Update 状態にある。供給側 S D F からの UpdateShadow オペレーションの受信により引き起こされる(E13)Shadow_Update_from_Supplier という外部イベントの受信後、Wait for Update Confirmation 状態へ遷移する。S D F は updateShadow アーギュメントの内容に従って UpdateShadow オペレーションを実行する。S D F が UpdateShadow オペレーションを完了させると、結果応答あるいはエラーが供給側 S D F に対して返される。S D F は Wait for Update 状態へ戻る。

17.132.3.1.1.2 DSAShadowBind と CoordinateShadowUpdate と共に受信した UpdateShadow

S D F 前条件 :

- (1) SDSM-ShCSi : Wait for Bind Result

S D F 後条件 :

- (1) SDSM-ShCSi : Wait for Update

S D F は最初、DSAShadowBind オペレーション以外に受信すべき他のオペレーションを待つ Wait for Bind Result 状態にある。CoordinateShadowUpdate オペレーション受信後の UpdateShadow オペレーション受信時、外部イベント(E3)Request_from_Supplier により、同じ状態へ遷移する。S D F は DSAShadowBind オペレーションを実行し、内部イベント(e5)SDF_Bind_Success により SDF Bound 状態へ遷移する。CoordinateShadowUpdate オペレーションを既に受信しているため、外部イベント(E7)Shadow_Coordinate_from_Supplier により Wait for Coordination

Result 状態へ遷移する。その後 S D F は CoordinateShadowUpdate オペレーションを実行し、内部イベント (e10)Shadow_Coordinate_Confirmed により Wait for Update 状態へ遷移する。UpdateShadow オペレーションを既に受信しているため、外部イベント(E13)Shadow_Update_from_Supplier により Wait for Update Confirmation 状態へ遷移する。最後に S D F は updateShadow アーギュメントの内容に従って UpdateShadow オペレーションを実行する。S D F が UpdateShadow オペレーションを完了させると、結果応答あるいはエラーが供給側 S D F に対して返される。S D F は Wait for Update 状態へ戻る。

17.132.3.1.2 消費側起動による UpdateShadow

S D F 前条件：

- (1) SDSM-ShCCi : Wait for Update

S D F 後条件：

- (1) SDSM-ShCCi : Wait for Update

S D F は最初 Wait for Update 状態にある。供給側 S D F からの UpdateShadow オペレーションの受信により引き起こされる(E14)Shadow_Update_from_Supplier という外部イベントの受信後、Wait for Update Confirmation 状態へ遷移する。S D F は updateShadow アーギュメントの内容に従って UpdateShadow オペレーションを実行する。S D F が UpdateShadow オペレーションを完了させると、結果応答あるいはエラーが供給側 S D F に対して返される。S D F は Wait for Update 状態へ戻る。

17.132.3.2 エラー処理

シャドウイングオペレーションに関連するエラーの一般的な扱いは I T U - T 勧告 X . 5 2 5 の 12 章に、オペレーションエラーを通知するための T C サービスは 18.1 節に記述されている。

18. 下位レイヤから想定されるサービス

18.1 TCから想定されるサービス

本勧告において定義されているSS7アプリケーションレイヤプロトコルは、一対のアプリケーションプロセス間の通信を提供するプロトコルである。SS7環境において、これはトランザクション機能を使用した一対のアプリケーションエンティティ(AE)間の通信として表現される。あるAEの機能は、アプリケーションサービスエレメント(ASE)のセットによって提供される。AE間の相互動作は、このASEによって提供されるサービスの利用によって示される。

アプリケーションコンテキスト(AC)が、1つの物理ノード内において、FEを区別するために使用されるのであれば、使用されるTCのバージョンはTCのダイアログ部をサポートしていなければならない。

この要求条件は、インターネットワーキングのために用いられているものだけではなく、全てのインタフェースに適用される。

表18-1/JT-Q1228は、どのTCバージョンが、定義されたINインタフェースをサポートするのに要求されている最低限のバージョンであるかを定義している：

表18-1/JT-Q1228* INAPインタフェースに対する最低限のTCの標準等の要求
(ITU-T Q.1228)

*

インタフェース	IN CS-2
SCF-SDF	TTC標準 JT-Q 771~Q774、および、ITU-T勧告 Q.775
SDF-SDF	TTC標準 JT-Q 771~Q774、および、ITU-T勧告 Q.775

18.1.1 共通手順

本節では、以降の節で定義されるように、特定のINAPインタフェースに対して、特定の手順とマッピング則がない場合に、INAPとTCとの間に適用される手順、及びマッピングを定義する。

18.1.1.1 正常手順

本節では、正常時にAE間でのメッセージ送信のために使用される手順と、TCプリミティブについて述べる。

INAPは、TCユーザとして、TCによって提供される構造ダイアログ機能のみを使用する。以下に示す状況は、2つの物理エンティティ間でメッセージが送信された時に起こる：

- ダイアログが確立される場合：TCユーザは、TC-開始要求プリミティブを発行する。
- ダイアログが継続される場合：TCユーザは、TC-継続要求プリミティブを発行する。
- ダイアログがもはや継続されない場合：TCユーザは、以下に示す条件に応じて、基本終了またはプリアレンド終了のいずれかによるTC-終了要求プリミティブを発行する。
 - 基本終了
 - ダイアログが確立された場合において、FEが、送信されたオペレーションに対してのエラー

または拒否コンポーネントの受信を期待しない場合には、FEによって、TC - 終了要求プリミティブ(基本)より、関係の終了をもたらすオペレーションが送出され得る。一旦FEダイアログリソースが解放されると、TTC標準JT-Q774に記述されているように、それらのオペレーションに対して受信したエラーまたは拒否コンポーネントは、TCにより破棄される。

- ダイアログが確立され、FEがダイアログの継続を期待しないで関係の終了を意味するオペレーションを受信した場合で、かつ、送るべきオペレーションが無い場合、そのFEから、コンポーネントを伴わないTC - 終了要求プリミティブ(基本)が送信され得る。
- プリアレンジド終了
あるエンティティが、送信された、関係の終了をもたらすオペレーションに対する応答として、エラーまたは拒否メッセージを期待する場合、そのダイアログは、最後の当該するオペレーションタイマの満了後、TC - 終了要求プリミティブ(プリアレンジド終了)で終了される。受信側エンティティは、これらのオペレーションの正常処理(すなわち、関係が終了された)後、TC - 終了要求プリミティブ(プリアレンジド終了)でダイアログを終了し得る。
- 一般に、プリアレンジド終了の使用は、通信している双方のエンティティが、相手側のエンティティがプリアレンジド終了を適用しているということを明らかに認識できる場合に限定されるべきである。他の全ての場合、基本終了が使用されるべきである。
- ダイアログが確立されない場合：クラス2あるいはクラス4オペレーションについてのみ、送信側TCユーザはTC - 開始要求プリミティブを発行し、オペレーションタイムアウト後、プリアレンジド終了によりローカルにそのダイアログを終了する。受信側TCユーザは、TC - 開始指示プリミティブを受信すると、ローカルにそのダイアログを終了する。

18.1.1.2 異常手順

本節では、AE間の異常状況を報告するために使用される手順とTCプリミティブについて記述する。エラーの場合については16章に定義される。

以下のプリミティブが、異常状況を報告するために使用される。

- INAPで定義されるオペレーションエラーは、TC - U - エラー要求プリミティブで報告される。
- TCユーザによるTCコンポーネントの拒否は、TC - U - 拒否要求プリミティブで報告される。
- エラーを検出、又はオペレーションを拒否するFEがTCダイアログの終了を決定した場合、エラー又は拒否を伴うTC - 終了要求プリミティブ(基本)を、そのTCダイアログの終了に使用できる。
- エラーを検出、又はオペレーションを拒否するSSF、SRF又はCUSFがダイアログ継続の可能性を認めた場合、エラー又は拒否を伴うTC - 継続要求プリミティブを、そのTCダイアログの継続に使用できる。 # # #
- ダイアログは、TCユーザによって、TC - U - アボート要求プリミティブで強制解放される。
- アプリケーションタイマ T_{SSF} 、 T_{SRF} 又は T_{CUSF} が満了した場合、TCダイアログが確立されたかどうかにかかわらず、ダイアログはアボート理由を伴うTC - U - アボートプリミティブを用いて終了される。 # # #

TCによって検出された異常状況のために、TC - U - 拒否要求プリミティブの送信についてはTC - R - 拒否指示プリミティブの受信と同様の、またTC - U - アボート要求プリミティブの送信についてはTC - P - アボート指示プリミティブの送信と同様の規定が適用される。

以下に示す規定が、異常状況のもとでTCダイアログを終了させるために適用される。

- アボート状態が検出され、TCダイアログが確立されている場合、TCダイアログは、アボート理由を伴うTC - U - アボートプリミティブにより終了される。

- アポート状態が検出され、TCダイアログが確立されていない場合、TCダイアログは、TC-U-アポートプリミティブによりローカルに終了される（アプリケーションタイム満了等の場合）。

エラー状況では、TCダイアログを終了させるためにプリアレンド終了は使用されない。いずれかのアプリケーションエンティティがエラー状況に遭遇した場合、可能ならば、相手エンティティには明示的にエラーが通知される。どのエンティティの視点からも遭遇したエラーが関係の終了を要求する場合、保留中のエラーコンポーネントまたは拒否コンポーネントが送信されるか否かにより、基本終了を伴うTC-終了要求プリミティブ、またはTC-U-アポート要求プリミティブでダイアログを終了する。

あるエンティティがTC-終了指示プリミティブを受信し、全てのコンポーネントが処理された後でFSMが関係を終了する状態にない場合、適切な内部エラーが与えられるべきである。

ダイアログの確立が完了する前に（TC-開始要求プリミティブに対する最初のTC指示プリミティブを、応答側エンティティから受信する前に）、そのダイアログが起動側エンティティによって終了されなければならない場合、TCユーザはプリアレンド終了を伴うTC-終了要求プリミティブ、またはTC-U-アポート要求プリミティブを発行する。これらのプリミティブの結果はローカルのみであり、このダイアログに対して受信する後続のTC指示は、TTC標準JT-Q774で規定されている異常手順に従って処理される。

18.1.1.3 ダイアログ処理

18.1.1.3.1 ダイアログ確立

INAPダイアログの確立は、3.1.3節で記述しているように、2つのアプリケーションプロセス、1つはダイアログ起動者（dialogue-initiator）、もう1つはダイアログ応答者（dialogue-responder）を含む。

アプリケーションコンテキストネゴシエーションは、全ての物理エンティティかつ/または全ての網において、サポートされていないかもしれない。

この手順は、以下の信号によって進められる。

- ダイアログ起動者からのTC-開始要求プリミティブ
- 応答側で発生するTC-開始指示プリミティブ
- 起動側で、または特定の条件下で発生する最初のTC-継続指示プリミティブ
 - 起動側で発生するTC-終了指示プリミティブ
 - 起動側で発生するTC-U-アポート指示プリミティブ
 - 起動側で発生するTC-P-アポート指示プリミティブ

TC-開始要求の送信

TC-開始要求プリミティブの発行前に、SACFは、アプリケーションコンテキスト名及び存在すればユーザ情報パラメータを蓄積する。

SACFは、TC-起動サービスを用いて、関連付けられるオペレーションの起動を要求する。起動手順の記述は18.1.1.4.1節を参照。

最後の起動要求の処理後、SACFはTC-開始要求プリミティブを発行する。

起動者SACFはこの後TC指示プリミティブを待ち、TC-U-アポート要求、またはプリアレンド終了に設定された解放手順パラメータを伴うTC-終了要求を除く他の要求は発行しない。

18.1.2.1.1節と18.1.2.1.2節に記述されている規定に従って、ダイアログを確立すべきでないためにTC指示プリミティブが期待されない場合、SACFは最後に関連付けられたTCオペレーションタイムが満了するのを待ち、プリアレンド終了に設定された解放手順パラメータを伴うTC-終了要求を発行する。

#

T C - 開始指示の受信

T C - 開始指示プリミティブを受信すると、応答者S A C Fは以下の処理を行う。

- アプリケーションコンテキスト名がそのプリミティブの中に含まれている場合、そのアプリケーションコンテキスト名を解析する。サポートされていれば、18.1.1.4.1 節に記述されているように、T C から受信した他の指示プリミティブを処理する。
- 18.1.2.1.1 節と 18.1.2.1.2 節に記述されている規定に従って
ダイアログを確立すべきでない場合、S A C FはT Cからの最終指示プリミティブを待ち、プリアレンジド終了に設定された解放手順パラメータを伴うT C - 終了要求を発行する。
- そのプリミティブに含まれるアプリケーションコンテキスト名がサポートされていない場合、T C - U - アポート要求プリミティブを発行する。代替のアプリケーションコンテキストが提供され得る場合、T C - U - アポート要求プリミティブの中にその名前が含まれる。

アプリケーションコンテキストネゴシエーションがT C - U - アポートプリミティブを用いることだけに制限されるかどうかは、今後の検討課題である。

#

最初のT C - 継続指示の受信

ダイアログに対する最初のT C - 継続指示プリミティブを受信すると、S A C Fはアプリケーションコンテキスト名パラメータの値をチェックする。この値がT C - 開始要求プリミティブの中で使われているものと一致する場合、S A C Fは 18.1.1.4.1 節に記述されているように、後続のT Cコンポーネント処理指示プリミティブを処理する。一致しない場合、S A C FはT C - U - アポート要求プリミティブを発行する。

アプリケーションコンテキストネゴシエーションがT C - U - アポートプリミティブを用いることだけに制限されるかどうかは、今後の検討課題である。

T C - 終了指示の受信

ダイアログが起動された状態においてT C - 終了指示プリミティブを受信すると、S A C Fはアプリケーションコンテキスト名パラメータの値をチェックする。この値がT C - 開始要求プリミティブの中で使われているものと一致する場合、S A C Fは 18.1.1.4.1 節に記述されているように、後続のT Cコンポーネント処理指示プリミティブを処理する。

T C - U - アポート指示の受信

T C - U - アポート指示プリミティブの受信は、ユーザアポート手順の一部として記述されている(18.1.1.3.4 節参照)。アポート理由が未サポートアプリケーションコンテキスト名である場合、応答側は、T C - U - アポート指示で代替となるアプリケーション名を提案するかもしれない。代替となるアプリケーションコンテキストが提案された場合、受信側エンティティはこの名前をチェックし、サポートできれば新しいダイアログが確立されてもよい。

T C - P - アポート指示の受信

T C - P - アポート指示プリミティブの受信は、プロバイダアポート手順の一部として記述されている(18.1.1.3.5 節参照)。

18.1.1.3.2 ダイアログ継続

一旦確立されると、そのダイアログは継続状態にある。

双方のアプリケーションプロセスは、どちらか一方がそのダイアログの終了を要求するまでI N A P A P D Uの転送を要求できる。

送信側エンティティ

SACFは、18.1.1.4.1 節に記述されているように、コンポーネント処理要求プリミティブを処理する。最終コンポーネント処理要求プリミティブの処理後、SACFはTC - 継続要求プリミティブを発行する。

受信側エンティティ

TC - 継続指示プリミティブを受信すると、SACFは0、1つ、またはいくつかのTCコンポーネント処理指示プリミティブを受け入れ、18.1.1.4.1 節に記述されるようにそれらを処理する。

18.1.1.3.3 ダイアログ終了

18.1.2.1.1 節及び 18.1.2.1.2 に記述される規則に従って
ダイアログを確立すべきでない場合、あるいはこれ以上ダイアログを維持すべきでない場合、ダイアログ起
動者及びダイアログ応答者の双方とも、ダイアログ確立後にそのダイアログの終了を要求できる。 #

ダイアログ終了手順は、以下のイベントによって進められる。

- TC - 終了要求プリミティブ
- TC - 終了指示プリミティブ

TC - 終了要求の送信

これ以上ダイアログを維持すべきでない場合、18.1.1.4.1 節に記述されているように、SACFは全てのコンポーネント処理要求プリミティブを処理する。

18.1.2.1.1 節と 18.1.2.1.2 節に記述されている規定に従って
最終コンポーネント処理要求プリミティブを処理した後、SACFは解放手順のパラメータを基本終了またはプリアレンド終了に設定し、TC - 終了要求プリミティブを発行する。 #
ダイアログを確立すべきでない場合は、18.1.1.3.1 節を参照。

TC - 終了指示の受信

TC - 終了指示プリミティブを受信すると、SACFは全てのコンポーネント処理指示プリミティブを受け入れ、18.1.1.4.1 節に記述されているようにそれらを処理する。

最終コンポーネント処理プリミティブを処理した後、ダイアログに関連する全てのリソースは解放される。

18.1.1.3.4 ユーザアボート

ダイアログ起動者とダイアログ応答者の双方とも、いつでもダイアログを強制解放する事ができる。

ユーザアボート手順は、以下のイベントの一つによって進められる。

- TC - U - アボート要求プリミティブ
- TC - U - アボート指示プリミティブ

TC - U - アボート要求の送信

TC - U - アボート要求プリミティブの発行後、ダイアログに関連する全てのリソースは解放される。

TC - U - アボート指示の受信

TC - U - アボート指示を受信すると、ダイアログに関連する全てのリソースは解放される。

18.1.1.3.5 プロバイダアポート

TCはダイアログ起動者側とダイアログ応答者側の双方においてダイアログを強制解放することができる。

プロバイダアポート手順は、以下のイベントによって進められる。

- TC - P - アポート指示プリミティブ

TC - P - アポート指示の受信

TC - P - アポート指示を受信すると、ダイアログに関連する全てのリソースは解放される。

18.1.1.3.6 TCダイアログプリミティブへのマッピング

TC - UNIサービスはINAPでは使用されない。

TCダイアログサービスへのパラメータマッピングは、以下の通りである。

TC - 開始サービスのパラメータの利用法は、以下の制限と共に 18.1.1.3.7 節に定義されている。

- TC - 開始サービスの着アドレスパラメータは、TC - 開始サービスに応答するAEのINAPアドレスに設定されるべきである。

注 - このパラメータで使用されるアドレスは、SCCPアドレス翻訳によって、代替AEのひとつにマッピングされてもよい。

- TC - 開始サービスのアプリケーションコンテキスト名パラメータは、起動AEと応答AE間で使用される特有のインタフェースに従って設定されるべきである。
- TC - 開始サービスの発アドレスパラメータは、TC - 開始サービスを起動するAEのINAPアドレスに明確に設定されるべきである。

TC - 継続サービスのパラメータの利用法は、以下の制限と共に 18.1.1.3.7 節に定義されている。

- TC - 継続サービスのアプリケーションコンテキスト名パラメータは、同じダイアログIDパラメータ値を持つTC - 開始サービスの、アプリケーションコンテキスト名パラメータの値が設定されるべきである。
- TC - 継続サービスの発アドレスパラメータは、存在するならTC - 継続サービスを起動するAEのINAPアドレスに明確に設定されるべきである。このパラメータはTC - 開始サービス後の、同じダイアログIDを持つ最初のTC - 継続サービスにのみ存在する。

TC - 終了サービスのパラメータの利用法は、以下の制限と共に 18.1.1.3.7 節に定義されている。

- TC - 終了サービスのアプリケーションコンテキスト名パラメータは、同じダイアログIDパラメータ値を持つTC - 開始サービスの、アプリケーションコンテキスト名パラメータの値が設定されるべきである。このパラメータは、TC - 開始サービスの直後にTC - 終了サービスが使用される時のみ存在する。

TC - U - アポートサービスのパラメータの利用法は、以下の制限と共に 18.1.1.3.7 節に定義されている。

- TC - U - アポートサービスのアポート理由パラメータは、TTC標準JT - Q 771で規定されるように使用されるべきである。
- TC - U - アポートサービスのアプリケーションコンテキスト名パラメータは、TC - 開始サービスで使用する値か、あるいは起動AEと応答AE間で確立するダイアログに使用できる代替値のどちらかが設定されるべきである。

注 - このパラメータは、TC - 開始指示へTC - U - アボートがすぐに応答する時のみ存在する。

TC - P - アボートサービスのパラメータの利用法は、以下の制限と共に 18.1.1.3.7 節に定義されている。

- TC - P - アボートサービスのP - アボートパラメータは、TCがダイアログを強制解放した理由を示すため、TCによって設定される。これは、TTC標準JT - Q 7 7 1に定義される値を取る。

18.1.1.3.7 TCダイアログパラメータへのデフォルトマッピング

ダイアログID

このパラメータの値は、インプリメント依存の方法でのINAP起動に関連付けられる。このパラメータは、INAP AEのリモートINAP AEへの特定TCダイアログが同一であることを証明する。

アプリケーションコンテキスト名

アプリケーションコンテキスト名パラメータは、TCダイアログによってサポートされるべきオペレーションセットに従って設定される。アプリケーションコンテキスト名の定義は7章から8章に記述されている。

ユーザ情報

このパラメータは、網オペレータ特有の方法で、起動側及び応答側アプリケーションプロセスの双方によって使用されてもよい。

コンポーネント表示

このパラメータは、TTC標準JT - Q 7 7 1で記述されているSACFによって使用される。

終了

TC - 終了要求プリミティブの解放手順パラメータの値は、18.1.2.1.1 節及び18.1.2.1.2 節に記述されている規則に従うSACFによって設定される。

#

サービス品質

TC要求プリミティブのサービス品質は、SACFによって以下の値へ設定される。

- 要求された順序制御
- リターンオプション。このパラメータは、インプリメント依存の方法でSACFによって設定される。

18.1.1.4 コンポーネント処理

18.1.1.4.1 INAPオペレーション手順

本節ではINAPオペレーション手順を記述する。

オペレーション起動

SACFは受信したパラメータからオペレーションアギュメントを構成し、TC - 起動手順を使用して、関連オペレーションの起動を要求する。プリミティブの中に関連IDが挿入されていれば、これは子オペレーションを示し、そのオペレーションが親オペレーションに関連することを暗示する。

オペレーション起動の受信

TC - 起動指示プリミティブを受信すると、SACFは以下の処理を行う。

- オペレーションコードが、アプリケーションコンテキストによってサポートされているオペレーション

と一致しない場合、適切な問題コード（認識不能オペレーション）を伴うTC-U-拒否要求プリミティブを使用して、拒否コンポーネントの転送を要求する。

- 関連IDが含まれていれば、以下の検証を実行する：関連IDによって参照されるオペレーションが関連オペレーションを許容しない場合、またはそのオペレーションコードが許容された関連オペレーションと一致しない場合、あるいは親オペレーション起動が実行中でない場合、適切な問題コード（期待されない関連応答、又は期待されない関連オペレーション）を伴うTC-U-拒否要求プリミティブを発行する。
- アーギュメントの型がオペレーションのために定義されたものでない場合、適切な問題コード（不正パラメータ）を伴うTC-U-拒否要求プリミティブを使用して、拒否コンポーネントの転送を要求する。
- INAPに関連するダイアログが解放されようとしているためにオペレーションが起動できない場合、問題コード（解放起動）を伴うTC-U-拒否要求プリミティブを使用して、拒否コンポーネントの転送を要求する。
- 要求されたオペレーションの実行に十分なINAP関連リソースが利用できない場合、問題コード（リソース限界）を伴うTC-U-拒否要求プリミティブを使用して、拒否コンポーネントの転送を要求する。
- その他の場合、TC-起動指示プリミティブを受け入れる。オペレーションがユーザ確認型の場合、SACFは相当する応答を待つ。

オペレーション応答

ユーザ確認型オペレーションのため、SACFは以下の処理を行う。

- クラス1またはクラス3オペレーションへの応答にエラー指示がない場合、受信パラメータから結果情報要素を構成し、TC-結果-Lサービスを使用して転送を要求する。
- クラス1またはクラス2オペレーションへの応答にエラー指示が含まれている場合、受信パラメータからエラーパラメータを構成し、TC-U-エラー要求プリミティブを使用して転送を要求する。

応答の受信

TC-結果-NL指示を受信すると、SACFは以下の処理を行う。

- 適切な問題コード（不正パラメータ）を伴うTC-U-拒否要求プリミティブを使用して、拒否コンポーネントの転送を要求する。

TC-結果-L指示を受信すると、SACFは以下の処理を行う。

- 結果パラメータの種別が、このオペレーションの結果のために定義されたものでない場合、適切な問題コード（不正パラメータ）を伴うTC-U-拒否要求プリミティブを使用して、拒否コンポーネントの転送を要求する。
- その他の場合、TC-結果-L指示プリミティブを受け入れる。

TC-U-エラー指示を受信すると、SACFは以下の処理を行う。

- エラーコードがSACFのために定義されていない場合、又は起動IDによって参照するオペレーションに関連付けられたものでない場合、適切な問題コード（認識不能エラーまたは期待されないエラー）を伴うTC-U-拒否要求プリミティブを使用して、拒否コンポーネントの転送を要求する。
- エラーパラメータの種別がこのエラーのために定義されたものでない場合、適切な問題コード（不正パラメータ）を伴うTC-U-拒否要求プリミティブを使用して、拒否コンポーネントの転送を要求する。
- その他の場合、TC-U-エラー指示プリミティブを受け入れる。

保留中のオペレーションに影響を与えるTC-U-拒否指示プリミティブを受信すると、SACFは以下

の処理を行う。

- TC - U - 拒否指示プリミティブを受け入れる。

問題コード(結果応答問題、期待されないエラー応答)を示すTC - L - 拒否を受信すると、SACFはアプリケーションプロセスに通知する。

問題コード(エラー応答問題、期待されないエラー応答)を示すTC - L - 拒否を受信すると、SACFはアプリケーションプロセスに通知する。

ローカルTCが、オペレーションへ影響を与えるプロトコルエラーを受信コンポーネント中に検出すると、このイベントが発生する。

問題コードが一般問題を示す時、起動IDがTCによって供給されていても、実行中オペレーションと関連付ける事はできないと考えられる。なぜなら、起動IDがローカル起動によるものかリモート起動によるものが明確でないからである。このような場合のSACFの動作は、本小節の「他のイベント」に記述されている。

TC - L - 取消指示を受信すると、SACFは以下の処理を行う。

- 関連付けられたオペレーションがクラス1オペレーションならば、アプリケーションプロセスに通知する。
- 関連付けられたオペレーションがクラス2オペレーションで、かつこのオペレーションのために定義された関連オペレーションがなければ、プリミティブを無視する。
- 関連付けられたオペレーションがクラス2オペレーションで、かつ関連オペレーションを持っているがどれも起動されていないならば、アプリケーションプロセスに通知する。
- 関連付けられたオペレーションがクラス2オペレーションで、かつこのオペレーションに回答する関連オペレーション起動を既に受信していれば、プリミティブを無視する。
- 関連付けられたオペレーションがクラス3オペレーションならば、アプリケーションプロセスに通知する。
- 関連付けられたオペレーションがクラス4オペレーションならば、プリミティブを無視する。

他のイベント

本小節は、どのオペレーションとも関連しない、または保留中のオペレーションに影響を与えないコンポーネント処理指示プリミティブを受信した時の、SACFの動作を記述する。

実行中オペレーションに影響を与えないTC - U - 拒否指示プリミティブ(すなわち結果応答問題、あるいはエラー応答問題を示す)を受信すると、

18.1.2.1.2 節に記述されている規則に従う

#

送信アプリケーションプロセスによってダイアログがまだ終了していなければ、アプリケーションプロセスはダイアログを強制解放、継続、あるいは終了することができる。これはまた、クラス4関連オペレーションに関連する、起動問題にもあてはまる。

実行中オペレーションに影響を与えないTC - R - 拒否指示プリミティブを受信すると(すなわち相手TCエンティティによりプロトコルエラーが検出されている時)、

18.1.2.1.2 節に記述されている規則に従う

#

送信アプリケーションプロセスによってダイアログがまだ終了していなければ、アプリケーションプロセスはダイアログを強制解放、継続、あるいは終了することができる。

実行中オペレーションと関連付けることができないTC - L - 拒否指示プリミティブを受信すると(すなわちローカルTCエンティティによりプロトコルエラーが検出されている時)、アプリケーションプロセスはダイアログを継続するか、終了して暗黙的に拒否コンポーネントの転送のきっかけを作るか、あるいはダイアログを強制解放することができる。

メッセージがネットワークレイヤにより配送できないことをSACFに通知するTC - 通知指示プリミティブを受信すると、アプリケーションプロセスはダイアログを終了するかリトライするかを決定する。このプリミティブは、リターンオプションが設定されているときのみ起こり得る（18.1.1.3.6節参照）。

18.1.1.4.2 TCコンポーネントプリミティブへのマッピング

TCコンポーネントサービスへのパラメータのマッピングは以下の通りである。

TC - U - 取消サービスは使用しない。

TC - 結果 - NLサービスは使用しない。

TC - 起動サービスのパラメータの利用法は、以下の制限と共に18.1.1.4.3節に定義される。

- TC - 起動サービスのオペレーションパラメータは、起動すべきINAPオペレーションの operation.&operationCode 値を含まなければならない。このオペレーションは、TCダイアログに対する合意済みのアプリケーションコンテキストによりサポートされる有効なオペレーションの1つであり、かつローカルAEにより起動可能でなければならない。
- TC - 起動サービスの「パラメータ」パラメータは、起動されるオペレーションのためにオペレーションパラメータにより規定された operation.&ArgumentType 値を含むべきである。

TC - 結果 - Lサービスのパラメータの利用法は、以下の制限と共に18.1.1.4.3節に定義される。

- TC - 結果 - Lサービスの起動IDパラメータは、結果が送信されるリモートAEからの、TC - 起動サービスの起動IDパラメータの値が設定されるべきである。
- TC - 結果 - Lサービスのオペレーションパラメータは、同じ起動IDパラメータ値を含むリモートAEからのTC - 起動サービスの、オペレーションパラメータの値が設定されるべきである。
- TC - 結果 - Lサービスの「パラメータ」パラメータは、オペレーションパラメータにより規定された、オペレーション結果の operation.&ResultType 値を含むべきである。

TC - U - エラーサービスのパラメータの利用法は、以下の制限と共に18.1.1.4.3節に定義される。

- TC - U - エラーサービスの起動IDパラメータは、エラーが送信されるリモートAEからのTC - 起動サービスの、起動IDパラメータの値が設定されなければならない。
- TC - U - エラーサービスのエラーパラメータは、送信すべきエラーの error.&errorCode の値が設定されるべきである。これは、operation.&Errors 仕様で定義された、起動オペレーションのために予期されるエラーの1つでなければならない。
- TC - U - エラーサービスの「パラメータ」パラメータは、エラーパラメータにより識別される、送信すべきエラーの error.&ParameterType の値が設定されるべきである。

TC - U - 拒否サービスのパラメータの利用法は、以下の制限と共に18.1.1.4.3節に定義される。

- TC - U - 拒否サービスの起動IDパラメータは、拒否されるリモートAEからのTCコンポーネントサービスの、起動IDパラメータの値が設定されるべきである。

TC - L - 取消サービスのパラメータの利用法は、18.1.1.4.3節に定義される。

18.1.1.4.3 TCコンポーネントパラメータへのデフォルトマッピング

起動ID

このパラメータは送信側アプリケーションプロセスによって設定される。これは、特定のTCダイアログ中のAEにより起動されるオペレーションのインスタンスの、唯一の同一性を表す。TCダイアログはダイアログIDパラメータにより識別される。

関連ID

このパラメータは送信側アプリケーションプロセスによって設定される。これは、ローカルAEによって

起動されたオペレーションと関連づけられる、特定のTCダイアログのリモートAEから受信したオペレーションの起動IDを表す。このパラメータは、リモートAEにより起動された元のオペレーションが関連オペレーションをもつ、と定義される時のみ存在する。起動されるローカルオペレーション種別は、関連付けられると定義されるオペレーションの1つと同じ種別でなければならない。

ダイアログID

このパラメータの値は、インプリメントに依存する方法でINAP起動に関連付けられる。これは、ローカルAEとリモートAE間のコンポーネントサービスを運ぶために確立したTCダイアログの同一性を表す。

クラス

このパラメータの値は、7章から8章のオペレーション定義に従って起動すべき、オペレーション種別に従って設定される。

タイムアウト

このパラメータの値は、起動したオペレーション種別に従って設定される。

最終コンポーネント

このパラメータは、TTC標準JT-Q771の記述に従って使用される。

問題コード

このパラメータは18.1.1.4.1節の記述に従って使用される。

アボート理由

このパラメータはSACFにより使用され、属性と符号化は網管理者により規定される。

18.1.2 S S F - S C F インタフェース	#
18.1.3 S C F - S R F インタフェース	#
18.1.4 S C F - C U S F インタフェース	#
18.1.5 S C F - S C F インタフェース	#

18.1.6 S C F - S D F インタフェース

Directory A S E により提供されるすべてのサービスは、単一のA Eに含まれる。トランザクション機能(T C)のコンポーネント処理部(C H A)はオペレーションの要求/応答の範例をサポートする。Directory A S E は、T C によって提供されるサービスへの、ディレクトリオペレーションパッケージの抽象構文記法のマッピング機能を提供する。T C のダイアログ処理部(D H A)は、対になったA E間のダイアログと呼ばれるアプリケーション連携の確立と解放を提供する。D U A とD S A 間のダイアログは、D U A によってのみ確立されてもよい。

18.1.6.1 正常手順

ダイアログ処理サービスは、DirectoryBind および DirectoryUnbind オペレーションをサポートするため、および Directory パッケージに含まれるオペレーションに関連付けられるA P D Uを送信するトリガとして使用される。

コンポーネントのグループ化は、TC - 開始及びTC - 継続サービスの適切な使用方法を通して、アプリケーションプロセス制御のもとに行われる。

TC - 終了サービスは unbind 手順をサポートするためだけに用いられる(すなわちコンポーネント送信のトリガには決して使用されない)。

空のTC - 継続要求プリミティブを受信すると、SDFはこのプリミティブを無視すべきである。

データベースへの要求を含んだTC - 終了要求を受信すると、SDFはデータベース要求を実行すべきでなく、要求されたTC - 終了サービスを unbind 手順としてみなさなければならない。ダイアログはその時終了される。

18.1.6.2 異常手順

SDFでTC - U - 拒否指示を受信すると、このプリミティブは無視されるべきである。

SDFでTC - R - 拒否指示を受信すると、ダイアログはTC - U - アポート要求と共に解放されるべきである。

SDFで拒否状態が検出されると、TC - 継続要求に続いてTC - U - 拒否要求が送信されるべきである。プリアレンド終了手順は決して使用されない。

アプリケーションプロセスは、TC - P - アポートサービス及びTC - 通知サービスの唯一のユーザである。

ダイアログでのTC - U - アポート指示又はTC - P - アポート指示の受信は、すべての要求処理を終了させる。要求された修正が発生したかどうかの確認は、アプリケーションプロセスの責任である。

適切なSDFにメッセージをルーティングするため、下位に存在するSCCPが使用できる着アドレスをTC - 開始要求プリミティブで提供する事は、アプリケーションプロセスの責任である。

18.1.6.3 ダイアログ処理

18.1.6.3.1 ダイアログ確立

18.1.6.3.2 ダイアログ継続

18.1.6.3.3 ダイアログ終了

18.1.6.3.4 ユーザアポート

SESEがアプリケーションコンテキストに含まれており、かつSE - U - アポートが要求される場合、SE - U - アポートはTC - U - アポートプリミティブにマッピングされる。 #

18.1.6.3.5 プロバイダアポート

SESEがアプリケーションコンテキストに含まれており、かつSE - P - アポートが要求される場合、SE - P - アポートはTC - P - アポートプリミティブにマッピングされる。 #

18.1.6.3.6 TCダイアログプリミティブへのマッピング

本節では、TTC標準JT - Q 7 7 1に定義されているTCダイアログ処理サービスへの、DirectoryBindおよびDirectoryUnbindサービスのマッピングを定義する。

Bind

DirectoryBind サービスは、以下のようにTCサービスにマッピングされる。

- a) TC - 開始サービスは、DirectoryBind オペレーション
やSETTransfer オペレーション

#

の起動に使用される。

- b) TC - 継続サービスは、DirectoryBind オペレーションの成功を報告するために使用される。
SESE がアプリケーションコンテキストに含まれている場合、TC - 継続サービスは 2 番目や 3 番目のやり取りに使用される。 #
- c) TC - U - アポートサービスは、DirectoryBind オペレーション
や SETransfer オペレーション #
の失敗を報告するために使用される。

これらのサービスにおけるパラメータの利用法は、以下の小節の中で制限される。

TC - 開始サービスのパラメータの利用法は、以下の制限と共に 18.1.1.3.6 節に定義される。

- TC - 開始サービスのアプリケーションコンテキスト名パラメータは、iNdirectoryAccessAC 又は inExtendedDirectoryAccessAC オブジェクトのアプリケーションコンテキスト名フィールドの値を取るべきである。
- TC - 継続サービスのダイアログ ID パラメータは、18.1.1.3.7 節に規定するように使用されるべきである。
Authorized Relationship Id は TC ダイアログ ID パラメータ上にマッピングできる。
- TC - 開始サービスのユーザ情報パラメータは、存在すれば直接参照フィールドと共に単一の EXTERNAL 型を含み、id-as-indirectoryBindingAS に設定され、符合化すべき値は DAPBinding-PDUs.bind.bind-invoke 型 (DirectoryBindArgument) の値でなければならない。
SESE がアプリケーションコンテキストに含まれていれば、TC - 開始サービスのユーザ情報パラメータは、seItem 型の値を含むべきである。 #

TC - 継続サービスのパラメータの利用法は、以下の制限と共に 18.1.1.3.6 節に定義される。

- TC - 継続サービスのダイアログ ID パラメータは、18.1.1.3.7 節に規定するように使用されるべきである。
Authorized Relationship Id は TC ダイアログ ID パラメータ上にマッピングできる。
- TC - 継続サービスのユーザ情報パラメータは、存在すれば直接参照フィールドと共に単一の EXTERNAL 型を含み、id-as-indirectoryBindingAS に設定され、符合化すべき値は DAPBinding-PDUs.bind.bind-result 型 (DirectoryBindResult) の値でなければならない。
SESE がアプリケーションコンテキストに含まれていれば、TC - 継続サービスのユーザ情報パラメータは seItem 型の値を含むべきである。 #

TC - U - アポートサービスのパラメータの利用法は、以下の制限と共に 18.1.1.3.6 節に定義される。

- TC - 継続サービスのダイアログ ID パラメータは、18.1.1.3.7 節に規定するように使用されるべきである。
Authorized Relationship Id は TC ダイアログ ID パラメータ上にマッピングできる。
- TC - U - アポートサービスのアプリケーションコンテキスト名パラメータは、TC - U - アポートが directoryBind オペレーションの失敗を報告していない限り、18.1.1.3.6 節に定義するように設定されるべきである。この場合、iNdirectoryAccessAC 又は inExtendedDirectoryAccessAC オブジェクトのアプリケーションコンテキスト名フィールドの値に設定される。
- TC - U - アポートサービスのユーザ情報パラメータは、TC - U - アポートが directoryBind オペレーションの失敗を報告していない限り、18.1.1.3.7 節に規定するように使用されるべきである。この場合、本パラメータは、存在すれば直接参照フィールドと共に単一の EXTERNAL 型を含み、id-as-indirectoryBindingAS に設定され、符合化すべき値は DAPBinding-PDUs.bind.bind-error 型 (DirectoryBindError) の値でなければならない。

Unbind

DirectoryUnbind サービスは、TC - 終了サービス上へマッピングされる。

TC - 終了サービスのパラメータの利用法は、以下の制限と共に 18.1.1.3.6 節に定義される。

- TC - 継続サービスのダイアログ ID パラメータは、18.1.1.3.7 節に規定するように使用されるべきである。

Authorized Relationship Id は、TC ダイアログ ID 上にマッピングできる。

18.1.6.4 コンポーネント処理

18.1.6.4.1 INAP オペレーション手順

DirectoryA S E は、アプリケーションプロセスによって利用される TC - L - 拒否および TC - L - 取消サービスを除き、TC コンポーネント処理サービスのユーザである。TC - L - 拒否指示の受信はアプリケーションプロセスにダイアログを放棄させる（すなわち、アプリケーションプロセスは TC - U - アポート要求プリミティブを発行する）。

TC - U - 取消サービスは決して使われることはない。

18.1.6.4.2 TC コンポーネントパラメータへのマッピング

DirectoryA S E サービスは、TC コンポーネント処理サービスにマッピングされる。オペレーションとエラーの TC サービスへのマッピングは、以下の制限つきで 18.1.1.4.2 節に定義される。

TC - 起動要求プリミティブのタイムアウトパラメータは、表 18 - 2 / JT - Q 1 2 2 8 に従って設定される。

表 18 - 2 / JT - Q 1 2 2 8 DAP オペレーションの TC タイマ値
(ITU - T Q . 1 2 2 8)

オペレーション	タイムアウト
search	中間値
modifyEntry	中間値
addEntry	中間値
removeEntry	中間値
execute	中間値

18.1.7 S D F - S D F インタフェース

18.1.7.1 正常手順

18.1.7.2 異常手順

18.1.7.3 ダイアログ処理

18.1.7.3.1 ダイアログ確立

18.1.7.3.2 ダイアログ継続

18.1.7.3.3 ダイアログ終了

18.1.7.3.4 ユーザアポート

もし、S E S E がアプリケーションコンテキストに含まれ、S E - U - アポートが要求されるのであれば、#
S E - U - アポートは T C - U - アポートプリミティブにマッピングされる。#

18.1.7.3.5 プロバイダアポート

もし、S E S E がアプリケーションコンテキストに含まれ、S E - P - アポートが要求されるのであれば、#
S E - P - アポートは T C - U - アポートプリミティブにマッピングされる。#

18.1.7.3.6 T C ダイアログプリミティブへのマッピング

D S P と D I S P は T C サービスにマッピングされ得る。本節では、DSABind, DSAUnbind, DSAShadowBind そして DSAShadowUnbind サービスの T T C 標準 J T - Q 7 7 1 に定義される T C ダイアログ処理サービスへのマッピングについて定義する。

DirectoryBind サービスは以下のように T C サービスにマッピングされる。

- a) T C - 開始サービスは DSAShadowBind と DSABind オペレーションの起動に使用される。
- b) T C - 継続サービスは DSAShadowBind と DSABind オペレーションの成功を報告するために使用される。
- c) T C - U - アポートサービスは DSAShadowBind と DSABind オペレーションの失敗を報告するために使用される。

パラメータの T C ダイアログサービスへのマッピングは以下の通り。

T C - 開始サービスのパラメータの利用は、以下の制限つきで 18.1.1.3.6 節に定義される。

- T C - 開始サービスのアプリケーションコンテキスト名パラメータは、inDirectorySystemAC もしくは shadowSupplierInitiatedAC もしくは shadowConsumerInitiatedAC オブジェクトの値をアプリケーションコンテキスト名フィールドに設定しなければならない。
- T C - 開始サービスのユーザ情報パラメータは、使用しているアプリケーションコンテキストの値に依存した EXTERNAL 型の値を含まなければならない。

もし、使用しているアプリケーションコンテキストが inDirectorySystemAC であるなら、ユーザ情報パラメータは、もしあれば、直接参照フィールドをもつ単一の EXTERNAL 型を含まなければならない、id-as-indirectoryDSABindingAS に設定しなければならない。そして符合化されるべき値は DSABinding-PDUs.bind.bind-invoke 型の値でなければならない。

もし、使用しているアプリケーションコンテキストが shadowSupplierInitiatedAC もしくは shadowConsumerInitiatedAC であるなら、ユーザ情報パラメータは、もしあれば、直接参照フィールドをもつ単一の EXTERNAL 型を含まなければならない、id-as-indisaShadowBindingAS に設定しなければならない。そして符合化されるべき値は DISPBindingPDUs.bind.bind-invoke 型の値でなければならない。

T C - 継続サービスのパラメータの利用は、以下の制限つきで 18.1.1.3.6 節に定義される。

- T C - 継続サービスのダイアログ I D パラメータは、T T C 標準 J T - Q 7 7 1 に指定されるよう使わ

れなければならない。認証された相互関係の ID は TC ダイアログ ID にマッピングされる。

- 最初の TC - 継続サービスのユーザ情報パラメータは使用しているアプリケーションコンテキストの値に依存した EXTERNAL 型の値を含まなければならない。

もし、使用しているアプリケーションコンテキストが inDirectorySystemAC であるなら、ユーザ情報パラメータは、もしあれば、直接参照フィールドをもつ単一の EXTERNAL 型を含まなければならない、id-as-indirectoryDSABindingAS に設定しなければならない。そして符合化されるべき値は DSABinding-PDUs.bind.bind-result 型の値でなければならない。

もし、使用しているアプリケーションコンテキストが shadowSupplierInitiatedAC もしくは shadowConsumerInitiatedAC であるなら、ユーザ情報パラメータは、もしあれば、直接参照フィールドをもつ単一の EXTERNAL 型を含まなければならない、id-as-indsaShadowBindingAS に設定しなければならない。そして符合化されるべき値は DISPBindingPDUs.bind.bind-result 型の値でなければならない。

TC - U - アポートサービスのパラメータの利用は、以下の制限つきで 18.1.1.3.6 節に定義される。

- TC - U - アポートサービスのダイアログ ID パラメータは、18.1.1.3.6 節に指定されるよう使われなければならない。認証された相互関係の ID は TC ダイアログ ID にマッピングされる。

- TC - U - アポートサービスのアプリケーションコンテキスト名パラメータは、TTC 標準 JT - Q 7 7 1 に指定されるよう使われなければならない。SDF が受信したアプリケーションコンテキスト名をサポートしていないことによってダイアログを拒否した場合、本パラメータは、inDirectorySystemAC もしくは shadowSupplierInitiatedAC もしくは shadowConsumerInitiatedAC オブジェクトの値をアプリケーションコンテキスト名フィールドに設定しなければならない。

- TC - U - アポートが bind オペレーションの失敗を報告しているとき、すなわち、アポート理由パラメータが「ダイアログ拒否」の値をもつ場合、TC - U - アポートサービスのユーザ情報パラメータは使用しているアプリケーションコンテキストの値に依存した EXTERNAL 型の値を含まなければならない。

もし、使用しているアプリケーションコンテキストが inDirectorySystemAC であるなら、ユーザ情報パラメータは、もしあれば、直接参照フィールドをもつ単一の EXTERNAL 型を含まなければならない、id-as-indirectoryDSABindingAS に設定しなければならない。そして符合化されるべき値は DSABinding-PDUs.bind.bind-error 型の値でなければならない。

もし、使用しているアプリケーションコンテキストが shadowSupplierInitiatedAC もしくは shadowConsumerInitiatedAC であるなら、ユーザ情報パラメータは、もしあれば、直接参照フィールドをもつ単一の EXTERNAL 型を含まなければならない、id-as-indsaShadowBindingAS に設定しなければならない。そして符合化されるべき値は DISPBindingPDUs.bind.bind-error 型の値でなければならない。

これら以外の場合は、パラメータなしでなければならない。

inDSAUnbind および inDSAShadowUnbind サービスは TC - 終了サービスにマッピングされる。TC - 終了サービスのパラメータの利用は、18.1.1.3.6 節に定義される。

SETtransfer サービスは以下を除き、他のサービスに対するのと同様の TC サービスにマッピングされる。 #

- もし SESE がアプリケーションコンテキストに含まれているならば、TC - 継続サービスのユーザ情報パラメータは seItem 型の値を含まなければならない。 #

18.1.7.4 コンポーネント処理

18.1.7.4.1 INAP オペレーション手順

18.1.7.4.2 TC コンポーネントパラメータへのマッピング

DirectoryASE サービスは TC コンポーネント処理サービスにマッピングされる。オペレーションとエラーの TC サービスへのマッピングは、18.1.1.4.2 節に下記の制限つきで定義される。

TC - 起動要求プリミティブのタイムアウトパラメータは、表 18 - 3 / JT - Q 1 2 2 8 に従い設定される。

表 18 - 3 / JT - Q 1 2 2 8 DSP / DISP の TC タイマ値
(ITU - T Q . 1 2 2 8)

オペレーション	タイムアウト
chainedAddEntry	中間値
chainedRemoveEntry	中間値
chainedModifyEntry	中間値
chainedExecute	中間値
chainedSearch	中間値
updateShadow	中間値
coordinateShadowUpdate	中間値
requestShadowUpdate	中間値

18.2 SCCP から想定されるサービス

本節では、SCF と SDF 間、SDF と SDF 間に適用される IN アプリケーションプロトコルのために、IN アプリケーションが利用し得る SCCP から要求されるサービスについて記述する。

*
*

記述されるサービスは、TTC 標準 JT - Q 7 1 1 ~ Q 7 1 4 と ITU - T 勧告 Q . 7 1 5 (SCCP ユーザガイド) に含まれている。これらの標準及び勧告では、バージョンの異なる SCCP 間での可能なインタワーキングならびにコンパチビリティについて加味されているべきある。

18.2.1 正常手順

SCCP は TC と MTP 間にリンクを形成し、MTP と連携して IN アプリケーションのためにネットワークサービスを提供する。提供されるネットワークサービスは、アプリケーションが下位レイヤに対して送信する信号メッセージを、対向するアプリケーションに正確に配送することを許容する。

18.2.2 SCCP からのサービス機能

18.2.2.1 SCCP コネクションレスサービス

以下に示すコネクションレスサービスが SCCP から想定されている。

- a) SCCP ユーザ間でシグナリングコネクションを可能とするネットワークアドレッシング
- b) 同一対地に対するメッセージストリームについて、SCCP ユーザが、「順序保証」あるいは「順序保証なし」オプションを起動可能とする順序制御
- c) 大きなユーザメッセージの分割あるいは再組み立て
- d) ルーティング、あるいは分割 / 再組み立ての失敗に起因して SCCP によって対向する SCCP ユーザに配送することのできないようなメッセージがあった場合に、SCCP ユーザが「エラー時メッセージ破棄」あるいは「エラー時メッセージ返却」を起動可能とするリターンオプション
- e) 輻輳制御

上述したようなサービスに用いられるプリミティブは以下の通り。

N - ユニットデータ要求プリミティブと N - ユニットデータ指示プリミティブは、それぞれデータ送信とデータ受信に用いられる。これらのプリミティブのパラメータには、着アドレス、発アドレス、順序制御、リターンオプション、ユーザデータが常時必須パラメータとして含まれている。

N - 通知指示プリミティブは、リターンオプションが設定されており、かつルーチングあるいは分割エラーが発生した場合、対向するユーザに配送できなかったデータを返送するために用いられる。

18.2.2.1.1 INAPアドレッシング

INAPアドレッシング要素は、SCCPで利用されることを想定しアプリケーションからTCに送信する、発アドレスあるいは着アドレスに含まれる情報から構成される。

アプリケーションは、SCCPが以下のいずれかによってメッセージをルーチングすることを想定している。

- a) 着信号局コード(DPC)、サブシステム番号(SSN)、MTPサービスアクセスポイント(SAP)を利用したルーチング
- b) グローバルタイトル(GT)と、オプションとしてSSN、DPC、MTP SAPを利用したルーチング

アプリケーションはSCCPに対して、発アドレスと着アドレスの両方について、SSN上のルーチングを利用するのかGT上のルーチングを利用するのかについても通知する。

INAPが付加的なアドレス情報を必要とする場合には、ルーチング形式の定まっていないGT部にそれを格納して配送する。

上述した方法a)は、着信号局コードと、着信号局コードに割り当てられたメッセージが配送される着SSNを知っている場合に利用される。ある国内網では、TTC標準JT-Q713に従って、例えばSCF機能にSSNを割り付けるように、異なる網特有アプリケーションに異なるSSNを割り付けていることもあり得る。

上述した方法b)は、GT内の要素の組み合わせによって示されるSCCPユーザにメッセージを配送する場合に利用される。この方法の利用の一例としては、異なる網間でのメッセージの配送が必要な場合が考えられる。この方法は、発側網が着側ネットワークの信号局コードとSSNの割り当てを知らないために利用される。メッセージが配送される終端ノードを決定する網は、着信号局コードとSSNを得るためにグローバルタイトルの翻訳を実施しなければならない。オプションとして発アドレスがSSNを含んでいた場合には、これは着SSNとして利用されるか、あるいは必要ならば適切な新たなSSNが提供される。着ノードが他の網でかつゲートウェイノードでなかった場合、アプリケーションは着側で利用されるSSNあるいはゼロでSSNを表現する。

INAPアドレッシングにGTが用いられた場合、INアプリケーションは、SCCPがTTC標準JT-Q713に定義されている以下に示すような要素をサポートしていることを想定する。

グローバルタイトル識別子

本識別子は、アドレス情報のフォーマットに利用される方法を表す。識別子の値である4は常に網間接続に利用される。

翻訳タイプ

翻訳タイプはTTC標準JT-Q713に定義されている。

番号計画

- 1) 網間インタフェースで利用されるアドレスについては、一般番号あるいはITU-T勧告E.164に従った番号計画でなくてはならない。
- 2) 他のインタフェースで利用されるアドレスについては、適当であると考えられるならば、TTC標準JT-Q713に定義されているどのような番号計画であってもよい。

グローバルタイトルアドレス情報 (GTAI)

これはアプリケーションが提供する実際のINAPアドレス情報であり、符号化スキームで示されるように符号化されている。

符号化スキーム

アプリケーションはGTAIのフォーマットに従って符号化スキームの値を設定する。許容される値はTTC標準JT-Q713に定義されている。

網プロバイダは、変換によるどのようなGTの値の変更によっても最初のGTの値に含まれるINAP特有情報を保護することを保証しなくてはならない。ネットワーク内のGT変換データは、GT情報が存在していた場合には、それをアドレス情報から削除してはならない。

これは網間接続のインタフェースに限らず、すべてのインタフェースに要求される。

発側ノードよりSSNでのルーチングがサポートされる場合には、国際網間動作にゼロ以外の国際標準SSNが要求される。

INAPサービス用に標準化されたゼロ以外のSSNがない場合は、発側ノードから着側ノードを含む網へはGTでのルーチングが必須となる。

INAPオペレーションをサポートするためのSCCPのバージョンは、少なくとも

92年版 White book でなくてはならない。

96年度版のTTC標準を使うことが望ましい。

*

18.2.2.1.2 順序制御

アプリケーションは、SCCPプロトコルクラス0あるいは1のいずれかを要求することを表明する。

クラス0は、ある着アドレスに対して、メッセージのシーケンシャルな配送を要求しない場合に利用される。

クラス1は、ある着アドレスに対して、メッセージのシーケンシャルな配送を要求する場合に利用される。

18.2.2.1.3 エラー返送

INアプリケーションは、SCCPによって対地にメッセージが配送されなかったことを知るためにエラー返送メカニズムを利用する。リターンオプションは、ルーチング、あるいは分割/再組み立ての失敗によって目的のユーザに配送されなかったことメッセージの返送を可能とする。

アプリケーションがリターンオプションを起動しメッセージが配送されなかった場合、SCCPはTTC標準JT-Q711に記述されているように「返送理由」を明確にする。N-通知指示プリミティブは発側ユーザに配送できなかったメッセージを返送するために用いられる。

18.2.2.1.4 分割/再組み立て

アプリケーションは、SCCPが一つのUDTメッセージで260オクテットまでのユーザデータ(アド

レス情報とTCメッセージを含む)を送信可能(分割と輻輳制御の実施のためXUDTメッセージでは248オクテットまでのユーザデータを送信可能)なため、長いユーザデータには分割で対応可能であることを想定している。

同様に、SCCPは、受信した分割メッセージの再組み立て機能を実行し、ユーザに対して再組み立てしたユーザデータを配送することが想定されている。

しかしながら、例えばSCCPで分割されるSCCPユーザデータとアドレスの理論上の最大値が3968オクテットであったとしても、SCCPユーザは、知っている最も大きなアドレスについては、許容される約2560オクテットまでに制限すべきである。また、アプリケーションは、TCメッセージに利用されるオクテットは2560オクテットまで許容としなくてはならない。

INアプリケーションはSCCPがユーザデータを16分割以上に分割することは想定しない。

18.2.2.1.5 輻輳制御

下位レイヤで発生し得る輻輳の制御を可能とするために、アプリケーションはメッセージの重要度を表す値を割り当てることができる。このパラメータの利用にはTTC標準JT-Q711~Q714が必要となる。

また、SCCP管理における以下に記述するような輻輳制御メカニズムが存在する。

18.2.2.2 SCCPコネクションオリエンテッドサービス

#

18.2.2.3 SCCP管理

#

19 . I Nの一般的なインタフェースセキュリティ	#
I Nの機能アーキテクチャ内のそれぞれのインタフェースは、インタフェースをまたがって引き渡される情報フローにセキュリティ機能を適用する必要があるかもしれない。本章では、それぞれのインタフェースが安全を保証された適切な通信を提供できるよう、I T U - T 勧告 X . 5 0 9 の認証の枠組み及び単純公開鍵 G S S A P I 機構 (S P K M) に基づく、セキュリティ機構と手順の一般的なセットを定義する。本章は、I N C S - 2 で利用する I T U - T 勧告 X . 5 0 9 のサブプロファイル及び S P K M を定義する。	#
I N C S - 2 に対するセキュリティ機能の提供は、S C F - S D F 及び S D F - S D F インタフェースに要求される。	#
19.1 インタフェースセキュリティ要求条件	#
I Nシステムでのセキュリティ要求条件は、以下の2つの主要な群に分割できる。	#
1) セキュリティフィーチャを提供するための要求条件 (網アクセスセキュリティの要求条件)	#
ユーザ / 端末認証 (サービスユーザが I Nシステムに自身の身元を証明する処理結果) やユーザプロファイル検証 (ユーザが機能の利用を承認されている事の検証) 等のセキュリティフィーチャを利用して、サービスへのユーザアクセス及びアクセスインタフェースにおける端末の網への攻撃 (例えばユーザなりすまし) の保護に関するさまざまな側面を包含する。	#
2) 網間接続インタフェースでの要求条件 (網間接続セキュリティ要求条件)	#
対向エンティティ間認証 (ある通信エンティティが網内の別のエンティティに自身の身元を証明する処理) や信号データ、もしくは T M N データの完全性、非拒絶性、機密性、エンティティプロファイル検証 (エンティティが機能の利用を承認されている事の検証) 等のセキュリティフィーチャを利用して、システムの内部インタフェースでの攻撃や装置内に格納されたデータに対しても、電気通信サービスの提供に関わる各種エンティティとエージェント間の相互動作の保護を包含する。	#
上記の問題のそれぞれに対して、インタフェースはデータ機密性、データ生成源認証及びデータ完全性を可能とする処理を提供しなければならない。これらの機能は、一般的に T T C 標準 J T - X 5 0 0 に記述される。	#
19.1.1 データ機密性	#
データ機密性は、関連部分のみがメッセージの内容を検査できることを保証する。メッセージを保護された形式に変換するアルゴリズムを適用する暗号は、十分なデータ機密性を提供する。ディレクトリとの全ての通信に対する暗号化方式は、ディレクトリセキュリティ機構もしくはディレクトリエントリ方式に適合すべきである。	#
19.1.2 データ完全性とデータ生成源認証	#
データ完全性は、メッセージの内容が転送中に改ざんされていないことを保証する。データ生成源認証は、メッセージが期待した生成源から来たことを保証する。メッセージの内容にハッシュ関数を適用した結果に暗号化アルゴリズムを適用したデジタル署名は、十分なデータ完全性とデータ生成源認証を提供する。ディレクトリとの全ての通信に対するデータ完全性の方法は、ディレクトリセキュリティ機構もしくはディレクトリエントリ方式に適合すべきである。	#
19.1.3 鍵管理	#
鍵管理は、暗号化システムで用いる秘密鍵の生成、分配および維持管理を参考にする。I T U - T 勧告	#

X . 5 0 9 に適合する鍵管理方式を使用すべきである。 #

19.2 手順とアルゴリズム

動作の焦点を絞るため、IN CS - 2 インタフェースのセキュリティ手順にいくつかの仮定が設けられるが、同時にセキュリティの許容レベルも与えられる。 #

- 一般的なセキュリティインタフェースは、可能な限りSPKMに準拠すべきである。 #
- 信号トラヒックの負荷や処理時間を軽減するよう、メッセージおよび手順内の個々のパラメータの数は最小化すべきである。 #
- 1種類のセキュリティのみをインタフェースに適用すべきである。選択の順序は、なし、ユーザ、対向である。 #
- ユーザと対向の両方の署名/暗号化が要求されたら、ユーザ署名/暗号化を使用すべきである。 #
- (メッセージ認証コード(MAC)手法を含む)暗号化とデジタル署名の機構のみを使用すべきである。 #

表 19 - 1 / JT - Q 1 2 2 8 # セキュリティレベルの分類 (ITU - T Q . 1 2 2 8) #

	署名	暗号化
なし	なし	なし
ユーザ	なし	ユーザ
	ユーザ	なし
	ユーザ	ユーザ
対向	なし	対向
	対向	なし
	対向	対向

19.2.1 認証手順

認証手順は、認証を確立する処理を定義し、かつ以下の機能を実現すべきである。 #

- FEへのアクセスが許容されているエンティティ(ユーザ/FE)を認証する。 #
- エンティティとFE間のすべてのメッセージに対して、セキュリティのレベルを確立する。 #
- 必要ならば、エンティティとFE間のメッセージの暗号化に対するセッション鍵を確立する。 #

現在のIN CS - 2の認証は、ITU - T勧告X . 5 0 9に定義された手順と共に、新たな手順であるSPKMを用いている。ITU - T勧告X . 5 0 9の主な目的は、ディレクトリサービスに対して単純認証、強力認証及びSPKMの機構を含む認証を提供することである。本節は、インプリメントの際にIN CS - 2の範囲内で考慮及びサポートすべきディレクトリ認証の外観を示す。 #

現在のIN CS - 2の認証手順は以下である。 #

- 1) IN CS - 1又はIN CS - 2に対する、パスワードを伴うBindオペレーションを用いた単純認証(ITU - T勧告X . 5 0 9の6.2節) #
- 2) IN CS - 2及びディレクトリに対する、強力な資格証明を伴うBindオペレーション、あるいはIN CS - 2に対するSPKMを用いた強力認証(ITU - T勧告X . 5 0 9の10章) #
- 3) IN CS - 1又はIN CS - 2に対する、Bindオペレーションを用いた外部手順 #

4) S P K M (単純公開鍵 G S S - A P I 機構)	#
1) S P K M は、安全なタイムスタンプを利用せずに達成すべき、片側のみ及び相互認証の両方を許容する。これは安全な認証にアクセスしなければならないにもかかわらず、安全な時刻にアクセスしなくともよい環境を可能とする。	#
2) S P K M は、通信しあう対向が (データ機密性、データ完全性などのために) 使用すべきさまざまなアルゴリズムを特定するため、アルゴリズム識別子を用いる。これは環境の多様性、将来の拡張性及び代替アルゴリズムに対して最大限の柔軟性をもたらす。	#
19.2.2 S P K M のアルゴリズムとネゴシエーション	#
S P K M についての付加情報を、本節で提供する。ディレクトリエントリ方式を含む他の認証機構に関する付加情報については、ユーザガイドおよび関連するディレクトリ仕様を参照のこと。	#
S P K M には、いくつかのアルゴリズム種別が採用されている。S P K M アルゴリズムでは、起動側及びターゲットという用語が使用されることに注意すべきである。I N C S - 2 では、ターゲットという用語は消費側として参照されてもよい。	#
19.2.2.1 データ機密性アルゴリズム (C - A L G)	#
暗号化されたデータの生成のために、対称アルゴリズムが用いられる。D E S - C B C アルゴリズムがデータ機密性に対する一例である。	#
19.2.2.2 データ完全性アルゴリズム (I - A L G)	#
このアルゴリズムの目的は、メッセージが正当な送信元によって組み立てられた後に何らかの手段で改ざんされていない事の保証である。使用されるアルゴリズムにより、このアルゴリズムのアプリケーションはメッセージに対して確実性も提供し、かつ (データ生成源認証を含む) 非拒絶性もサポートする。m d 5 W i t h R S A、D E S - M A C、m d 5 - D E S - C B C および s u m 6 4 - D E S - C B C のようなアルゴリズムが、データ完全性に対する例である。	#
19.2.2.3 鍵管理アルゴリズム (K - A L G)	#
このアルゴリズムの目的は、起動側及び応答側 (ターゲットと呼ばれる) の双方で使用する対称鍵 (コンテキスト鍵と呼ばれる) を、確立したコンテキスト上で確立することにある。C - A L G に使用する鍵及び鍵化された何らかの I - A L G (例えば D E S - M A C) は、このコンテキスト鍵から得られる。鍵の確立は I T U - T 勧告 X . 5 0 9 の認証交換によって行われ、その結果として共有対称鍵が認証される。R S A E n c r i p t i o n や d h K e y A g r e e m e n t のようなアルゴリズムが鍵管理に対する例である。	#
19.2.2.4 副鍵導出アルゴリズムに対する 1 ウェイ機能 (O - A L G)	#
取り決めた K - A L G を用いて確立したコンテキスト鍵がある場合、起動側とターゲットはさまざまな C - A L G や鍵化された I - A L G に対する副鍵のセットを得ることができなければならない。	#
19.2.2.5 S P K M ネゴシエーション	#
S P K M におけるコンテキストの確立の間、起動側は利用可能な機密性アルゴリズムのセットと利用可能な完全性アルゴリズムのセットをターゲットに提示する (完全性アルゴリズムという用語はデジタル署名アルゴリズムを含む)。ターゲットが選択した機密性アルゴリズムは、確立したコンテキスト上で C - A L G に使用されてもよいものとなり、ターゲットが選択した完全性アルゴリズムは、確立したコンテ	#

キスト上で I - A L G に使用されてもよいものとなる (ターゲットは、提示されたそれぞれのリストがサ
ポートするサブセットを同じ相対順序で返送することでアルゴリズムを選択する)。コンテキスト上のど
のメッセージに、どんな C - A L G と I - A L G が使用されてもよく、合意したセットの最初の機密性ア
ルゴリズムと最初の完全性アルゴリズムが、そのコンテキストのデフォルトアルゴリズムとなる。 #

特定のコンテキストに対して合意された機密性及び完全性アルゴリズムは、保護品質 (Q O P) パラメ
ータの正当な値を定義する。片側認証において応答がターゲットから期待されないなら、起動側に提示さ
れたアルゴリズムがコンテキスト上で使われ得るものである (これがターゲットにとって非許容であれば、
このコンテキストでのアソシエーションの確立は成功しない)。 #

さらに、最初のコンテキスト確立トークン内で、起動側はセット内の最初のアルゴリズム (推奨アルゴ
リズム) に対応した鍵 (もしくは部分鍵) とともに K - A L G のセットを提示する。この K - A L G がタ
ーゲットにとって非許容であれば、ターゲットはセット内の他の K - A L G の 1 つを選択し、応答内でそ
の選択に対応した鍵 (もしくは部分鍵) とともにその選択を送信しなければならない (さもなければ、こ
のコンテキストでのアソシエーションは確立しない)。必要であれば (すなわち、ターゲットが d h K e
y A g r e e m e n t のような 2-pass K - A L G を選択すれば)、起動側はターゲットへの応答内にその
部分鍵を送信する。 #

最後に、最初のコンテキスト確立トークン内で、起動側は O - A L G (副鍵導出アルゴリズム) のセッ
ト (応答が期待されないなら単一の O - A L G) を提示する。ターゲットに選択された (単一の) O - A
L G が、コンテキスト上で使われる副鍵導出アルゴリズムとなる。 #

19.2.3 3ウェイ相互認証

2つの機能エンティティ (F E) 間の安全なアソシエーションの確立における (S P K M を含む) 3方
向相互認証手順の利用は、I T U - T 勧告 X . 5 1 9 に記述されるように、(I T U - T 勧告 X . 8 3 1
および X . 8 3 2 に定義された) 一般的な上位レイヤセキュリティ (G U L S) セキュリティ交換サービ
ス要素 (S E S E) プロトコルの利用を意味する。 #

19.2.4 資格証明の割り当て

ユーザもしくは F E への資格証明割り当て手順は、一般的に T T C 標準 J T - X 5 0 0 に定義されてい
る。 #

19.3 トークンへのセキュリティ情報フロー定義のマッピング

S C F - S D F インタフェースの場合、コンテキスト確立トークンは Bind 手順で運ばれ、メッセージ毎
のトークンは D A P オペレーションで使われる。 #

S D F - S D F インタフェースの場合、コンテキスト確立トークンは Bind 手順で運ばれ、メッセージ毎
のトークンは D S P および D I S P オペレーションで使われる。 #

19.4 セキュリティ F S M 定義

ここに記述されるセキュリティ F S M は一般的な機構である。これらは、2つの F E 間のインタフェ
ースをまたがってセキュリティ手順がどのように扱われるかを示す。 #

19.4.1 2ウェイ相互認証 F S M

19.4.1.1 出側 F S M

出側 F S M はインタフェースの発側を制御する。これを図 1 9 - 1 / J T - Q 1 2 2 8 に示す。 #

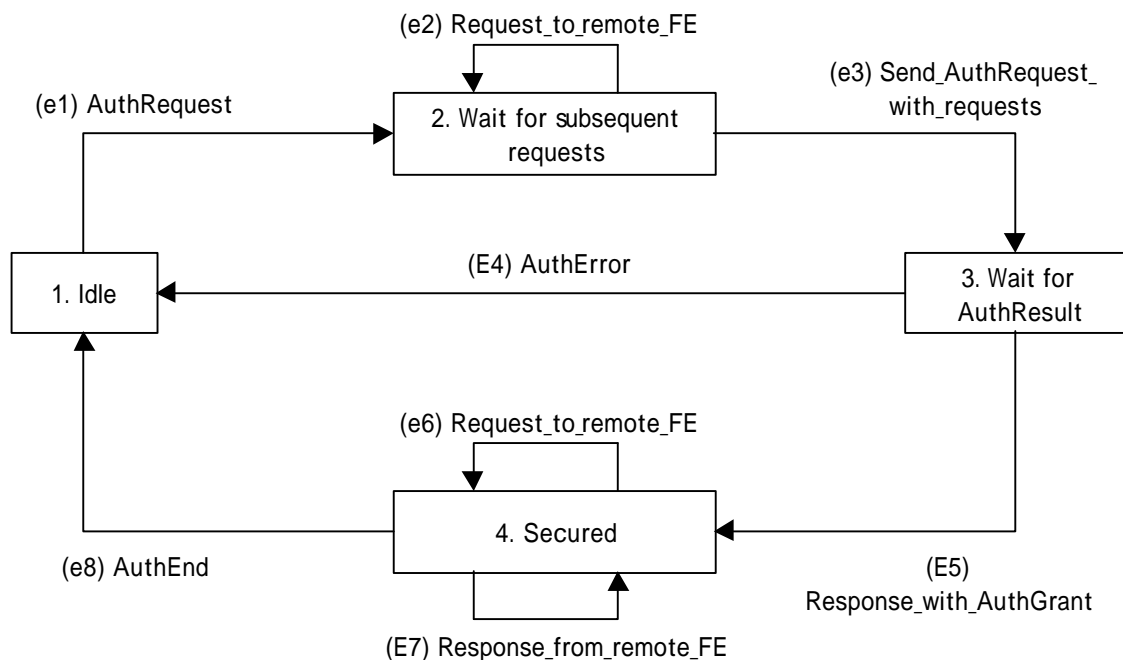


図 19 - 1 / J T - Q 1 2 2 8 # 発側 F S M 状態
(I T U - T Q . 1 2 2 8)

19.4.1.1.1 状態 1 : Idle

この状態では、以下のイベントが考えられる。

- (e1) AuthRequest : これは内部イベントであり、サービス機能にアクセスを開始するため、サービス論理がリモート F E とのアソシエーションの生成を必要とすることでもたらされる。このイベントは状態 2、Wait for subsequent requests への遷移をもたらす。

19.4.1.1.2 状態 2 : Wait for subsequent requests

この状態では、AuthRequest オペレーションとともに (同じメッセージで) リモート F E に送信すべき後続オペレーションが期待される。この状態では、以下の 2 つのイベントが考えられる。

- (e2) Request_to_remote_FE : これはオペレーションの受信により引き起こされる内部イベントである。オペレーションはデリミタの受信 (もしくはタイマ満了) までバッファリングされる。発側 F S M は同じ状態に留まる。
- (e3) Send_AuthRequest_with_requests : これは送信すべき最後のオペレーションの受信を示すデリミタの受信によって引き起こされる内部イベントである。いったんデリミタを受信すると、AuthRequest オペレーションのアーギュメントと、もしあれば他のオペレーションのアーギュメントを含んだメッセージがリモート F E に送信される。このイベントは、この状態から状態 3、Wait for AuthResult への遷移を生じる。

19.4.1.1.3 状態 3 : Wait for AuthResult

この状態では、ローカル F E はリモート F E からの応答を待っている。この状態では、2 つのイベントが考えられる。

- (E4) AuthError : これは外部イベントであり、既にリモート F E に発行した AuthRequest オペレーションへのエラーを受信することにより生じる。このイベントは、この状態から状態 1、Idle

への遷移を生じる。

- (E5) Response_with_AuthGrant : これは外部イベントであり、もしあれば既にリモート F E に発行した他のオペレーションへの結果と結合された、AuthRequest 結果の受信により引き起こされる。このイベントは状態 4、Secured への遷移をもたらす。

19.4.1.1.4 状態 4 : Secured

この状態では、ローカル F E はリモート F E への認証されたアクセスを確立しており、サービス論理からリモート F E への要求を待っているか、もしくは既にリモート F E に発行したオペレーションに対する応答を待っている。この状態では、3つのイベントが考えられる。

- (e6) Request_to_remote_FE : これは内部イベントであり、サービス論理がリモート F E にアクセスする必要が生じた時に引き起こされる。発側 F S M は同じ状態に留まる。
- (E7) Response_from_remote_FE : これは外部イベントであり、既にリモート F E に発行したオペレーションに対する応答の受信により引き起こされる。発側 F S M は同じ状態に留まる。
- (e8) AuthEnd : これは内部イベントであり、サービス論理がリモート F E への認証されたアクセスを終了する必要が生じた時に引き起こされる。このイベントは状態 1、Idle への遷移を生じる。

19.4.1.2 入側 F S M

入側 F S M はインタフェースの着側を制御する。これを図 19 - 2 / J T - Q 1 2 2 8 に示す。

各状態は以降の各節で議論される。

2つ以上の状態に適用可能な一般則は以下の通りである。

いかなる状態においても、リモート F E との対話が終了されれば、着側 F S M は呼に割り当てられていたすべてのリソースが割り当て解除されることを保証した後に、Idle 状態に戻る。

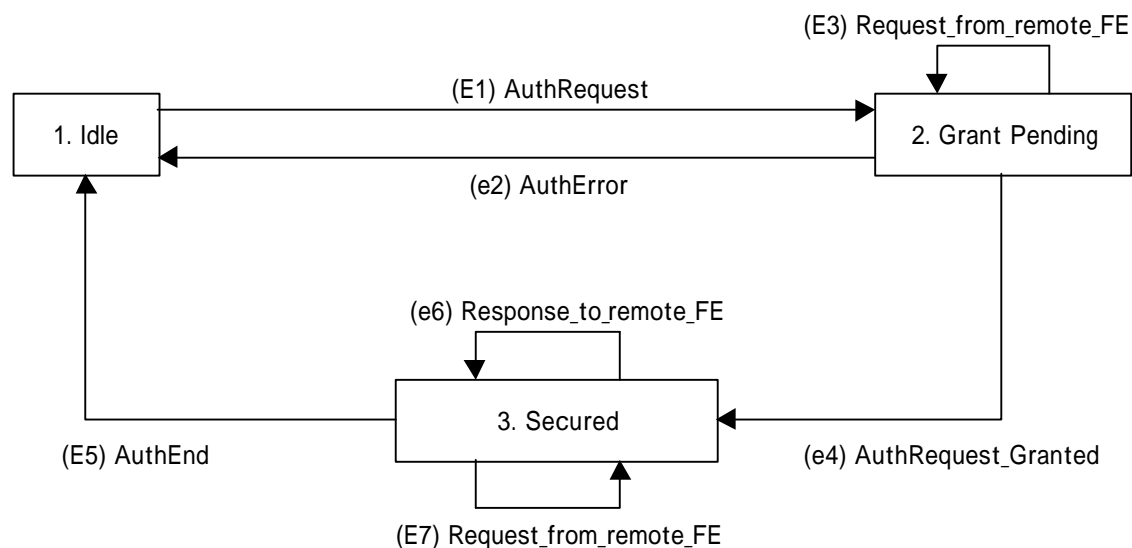


図 19 - 2 / J T - Q 1 2 2 8 # 着側 F S M 状態 (I T U - T Q . 1 2 2 8)

19.4.1.2.1 状態 1 : Idle

この状態で唯一許容されるイベントは以下である。

- (E1) AuthRequest : これは外部イベントであり、AuthRequest オペレーションをリモート F E から受

信することにより生じる。このイベントは、この状態から状態 2、Grant Pending への遷移をもたらす。 #

19.4.1.2.2 状態 2 : Grant Pending

この状態では、AuthRequest はリモート FE から受信済みである。ローカル FE は認証手順を実行中である。この状態では、3つのイベントが考えられる。 #

(e2) AuthError : これは内部イベントであり、ローカル FE に既に発行した AuthRequest オペレーションの失敗により引き起こされる。このイベントは、この状態から状態 1、Idle への遷移をもたらす、AuthError が起動側 SCF に返送される。 #

(E3) Request_from_remote_FE : これは外部イベントであり、AuthRequest オペレーションからの結果が決定する前のオペレーションの受信により引き起こされる。着側 FSM は同じ状態に留まる。 #

(e4) AuthRequest_Granted : これは内部イベントであり、ローカル FE に既に発行した AuthRequest オペレーションの成功によりもたらされる。このイベントは、この状態から状態 3、Secured への遷移を引き起こす。 #

19.4.1.2.3 状態 3 : Secured

この状態では、リモート FE のローカル FE へのアクセスが認証され、リモート FE からくるオペレーションが受理される。リモート FE からの要求を待つとともに、ローカル FE は既に発行されたオペレーションへの応答を送信することができる。この状態では、3つのイベントが考えられる。 #

(E5) AuthEnd : これは外部イベントであり、AuthEnd オペレーションをリモート FE から受信することにより引き起こされる。認証されたアソシエーションは終了され、関連する全てのリソースが解放される。このイベントはこの状態から状態 1、Idle への遷移をもたらす。 #

(e6) Response_to_remote_FE : これは内部イベントであり、以前にリモート FE によって発行されたオペレーションが完了することにより生じる。これらのオペレーションの結果もしくはエラーがリモート FE に返送される。着側 FSM は同じ状態に留まる。 #

(E7) Request_from_remote_FE : これは外部イベントであり、リモート FE から要求を受信することにより生じる。着側 FSM は同じ状態に留まる。 #

19.4.2 3ウェイ相互認証 FSM

3ウェイ相互認証のための、出側及び入側の一般的なセキュリティ FSM が本小節に記述される。ここで記述されるセキュリティ FSM は一般的な機構である。これらは、2つの FE 間のインタフェースをまたがって、セキュリティ手順がどのように処理されるかを示す。 #

19.4.2.1 3ウェイ相互認証の出側 FSM

出側 FSM はインタフェースの発側を制御する。これを図 19 - 3 / JT - Q 1 2 2 8 に示す。 #

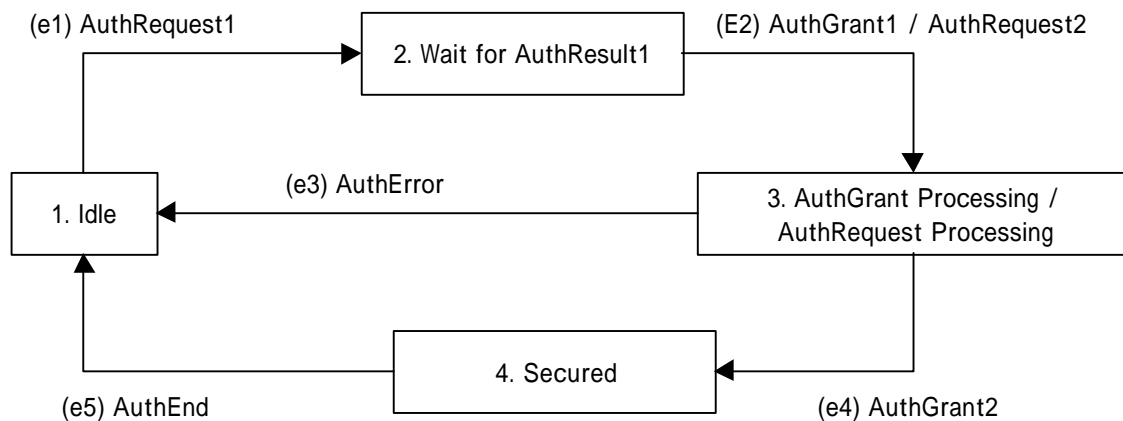


図 19 - 3 / J T - Q 1 2 2 8 # 3ウェイ相互認証の発側 F S M状態
(I T U - T Q . 1 2 2 8)

19.4.2.1.1 状態 1 : Idle

この状態では、以下のイベントが考えられる。

- (e1) AuthRequest1 : これはサービス機能へのアクセスを開始するために、サービス論理がリモート F E とのアソシエーションの生成を必要とする場合にもたらされる内部イベントである。このイベントは状態 2、Wait for AuthResult1 への遷移を生じる。

19.4.2.1.2 状態 2 : Wait for AuthResult1

この状態では、リモート F E からの最初のチャレンジ (AuthRequest1 内で送信される) に対する応答が期待されている。この状態では、以下のイベントが考えられる。

- (E2) AuthGrant1/AuthRequest2 : これは外部イベントであり、リモート F E からの最初のチャレンジおよび後続のチャレンジに対する応答の受信によりもたらされる。このイベントは、この状態から状態 3、AuthGrant Processing/AuthRequest Processing への遷移をもたらす。

19.4.2.1.3 状態 3 : AuthGrant Processing/AuthRequest Processing

この状態では、ローカル F E は AuthRequest1 への正しい応答を確認する。加えて、AuthRequest2 への適切な応答が生成される。この状態では、2つのイベントが考えられる。

- (e3) AuthError : これは内部イベントであり、AuthRequest1 への応答としてリモート F E から受信した AuthGrant1 の処理中のエラーによってもたらされる。このイベントは、この状態から状態 1、Idle への遷移をもたらす。
- (e4) AuthGrant2 : これは内部イベントであり、リモート F E に発行した AuthRequest2 への適切な応答の処理によってもたらされる。このイベントは状態 4、Secured への遷移をもたらす。

19.4.2.1.4 状態 4 : Secured

この状態では、ローカル F E はリモート F E との片側認証を確立している。リモート F E がいったん Secured 状態に進むと、リモート F E からの要求が処理されてもよい (これらのイベントは明示されていない)。この状態では1つのイベントが考えられる。

- (e5) AuthEnd : これは内部イベントであり、サービス論理がリモート F E への認証されたアクセスの終了を必要とすることによりもたらされる。このイベントは状態 1、Idle への遷移をもたらす。

す。

#

19.4.2.2 3ウェイ相互認証の入側 F S M

#

入側 F S M はインタフェースの着側を制御する。これを図 19 - 4 / J T - Q 1 2 2 8 に示す。

#

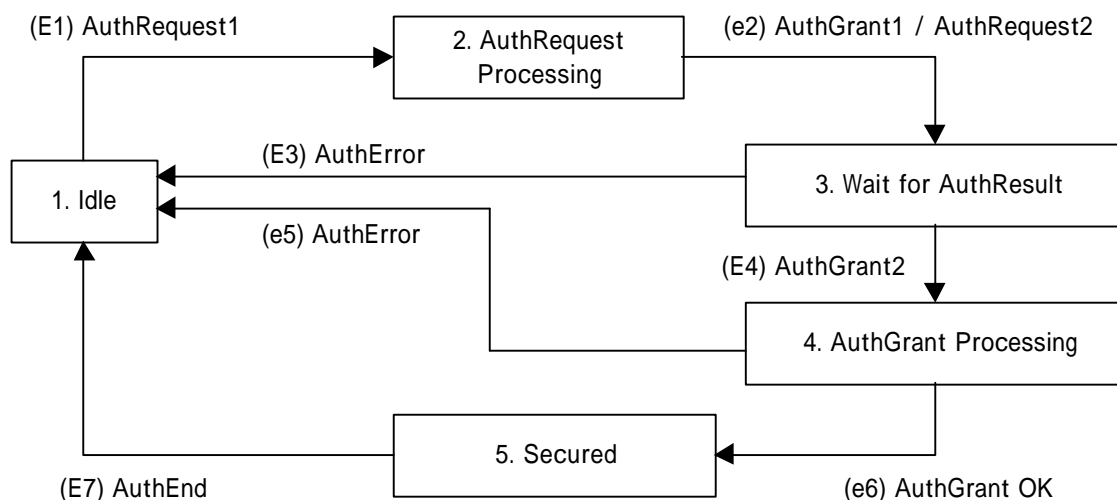


図 19 - 4 / J T - Q 1 2 2 8 # 3ウェイ相互認証の着側 F S M 状態
(I T U - T Q . 1 2 2 8)

#

#

19.4.2.2.1 状態 1 : Idle

#

この状態で唯一許容されるイベントは以下である。

#

(E1) AuthRequest1 : これは外部イベントであり、リモート F E からの AuthRequest1 情報の受信によりもたらされる。このイベントは、この状態から状態 2、AuthRequest Processing への遷移をもたらす。

#

#

#

19.4.2.2.2 状態 2 : AuthRequest Processing

#

この状態では、リモート F E から AuthRequest を既に受信している。ローカル F E は認証手順を実行している。この状態では 1 つのイベントが考えられる。

#

#

(e2) AuthGrant1/AuthRequest2 : これは内部イベントであり、ローカル F E へ以前に発行した AuthRequest1 オペレーションの完了により引き起こされる。加えて、ローカル F E の認証チャレンジ (AuthRequest2) が生成される。このイベントは、この状態から状態 3、Wait for AuthResult への遷移を生じる。

#

#

#

#

19.4.2.2.3 状態 3 : Wait for AuthResult

#

この状態では、ローカル F E はリモート F E からの確認、つまり AuthRequest1 チャレンジが受け入れられた事に応答する AuthGrant1 か、AuthRequest2 チャレンジが受信された事への応答 (AuthGrant2) のどちらかを待つ。この状態では、2 つの外部イベントが考えられる。

#

#

#

(E3) AuthError : これは外部エラーイベントであり、そのチャレンジに対する無効な応答を受信した時に、リモート F E によって生成される。このイベントは、この状態から状態 1、Idle への遷移をもたらす。

#

#

#

(E4) AuthGrant2 : これは外部イベントであり、ローカル F E チャレンジ (AuthRequest2) に対する

#

応答 (AuthGrant2) を含めて、リモート F E によって生成される。このイベントは、この状態から状態 4、AuthGrant Processing への遷移をもたらす。 #

19.4.2.2.4 状態 4 : AuthGrant Processing

この状態では、ローカル F E はチャレンジ (AuthRequest2) に対して受信した応答 (AuthGrant2) を確認する。この状態では、2 つの内部イベントが考えられる。 #

(e5) AuthError : これは内部イベントであり、そのチャレンジに対する無効な応答を受信した時に、ローカル F E によって生成される。このイベントは、この状態から状態 1、Idle への遷移をもたらす。 #

(e6) AuthGrant OK : これは内部イベントであり、そのチャレンジに対する有効な応答を受信した時に、ローカル F E によって生成される。このイベントは状態 5、Secured への遷移をもたらす。 #

19.4.2.2.5 状態 5 : Secured

この状態では、ローカル F E はリモート F E との間に片側認証を確立している。リモート F E も Secured の状態にあるため、リモート F E からの要求が処理されてもよい (これらのイベントは明示されていない)。この状態では 1 つのイベントが考えられる。 #

(E7) AuthEnd : これは外部イベントであり、サービス論理がリモート F E への認証されたアクセスの終了を必要とするにより生じる。このイベントは状態 1、Idle への遷移をもたらす。 #

付属資料A	INAPのSDL図	#
付属資料B	インテリジェントネットワーク定義で使われた用語解説	*
1 . 用語と定義		*
	この標準の目的のために、次の定義が適用される :	*
	アクセス : ユーザとネットワーク間の相互動作手段。	*
	アプリケーションエンティティ (AE) : システムに依存しないアプリケーション動作セットであり、アプリケーションエージェントによりアプリケーションサービスとして利用できる。例えば、1つのアプリケーション処理の通信面の一部あるいはすべてを担うアプリケーションサービス要素のセットがこれにあたる。	*
	アプリケーションエンティティインボケーション (AEI) : AEタイプを構成する通信機能のすべてあるいはサブセットを実行する実際の「実行時間」プログラム。	*
	アプリケーションコンテキスト (AC) : X . 207 参照。	*
	アプリケーションコンテキストネゴシエーション (ACN) : ユーザプロトコルASEsのセットで交換されるトランザクションの開始時に確立されるネゴシエーション処理。	*
	アプリケーションサービス要素 (ASE) : X . 207 参照。	*
	アプリケーションプロトコルデータユニット (APDU) : アプリケーションレイヤ制御情報及びデータを処理するPDU。	*
	インタフェース : 共有境界線、例えば2つのサブシステムあるいは2つのデバイス間の境界線。	*
	インテリジェントネットワーク (IN) : 顧客管理下のものも含めた新しい能力やサービスを導入するための設備を柔軟に提供することができるテレコミュニケーションネットワークのアーキテクチャ。	*
	インテリジェントネットワークアプリケーションプロトコル (INAP) : インテリジェントネットワークアプリケーションのためのプロトコル。	*
	エンティティ : 個々に考慮される部分、デバイス、サブシステム、機能ユニット、装置、システム。	*
	開放型システム間相互接続 (OSI) : ITU - T 勧告 X . 200 の基本参照モデルを参照。	*
	機能 : ある特定の目的のために定義された処理のセット。	*
	機能エンティティアクセスマネージャ (FEAM) : 機能エンティティが特定状態にいる間、特定ステータスの結果として機能エンティティに実行される動作の管理機能	*

公衆交換電話網 (P S T N) : 公衆加入者に電話サービスを提供するためにつくられた通信網。	*
呼制御エージェント機能 (C C A F) : ユーザにネットワークアクセス機能を提供する機能エンティティ。 サービスを提供する際に呼制御機能エンティティと相互動作する。	* *
呼制御機能 (C C F) : 呼 / コネクション処理と制御を提供するアプリケーションプロセス。	*
サービスアクセスポイント (S A P) : 各種サービスの動作を実現する物理エンティティ	*
サービス交換ポイント (S S P) : サービス交換機能を実装する物理エンティティ。	*
サービス交換・制御ポイント (S S C P) : サービス制御機能、サービスデータ機能とサービス交換 / 呼制御機能を含む物理エンティティ。	* *
サービス制御機能 (S C F) : インテリジェントネットワークサービスにより提供されるサービスにおいて、機能エンティティを制御するサービス論理のアプリケーション。	* *
サービス制御ポイント (S C P) : サービス制御機能を実装するインテリジェントネットワーク内の物理エンティティ。	* *
サービスデータ機能 (S D F) : サービスデータのテンプレートに準じてサービスデータ管理を行う機能のセット。	* *
サービスデータポイント (S D P) : サービスデータ機能を実装する物理エンティティ。	*
サービス統合デジタル網 (I S D N) : I T U - T 勧告 I . 1 1 2 の 2.3 節を参照。	*
サービス論理処理プログラム (S L P) : サービス論理を含むソフトウェアプログラム。	*
情報フロー : 機能エンティティ通信間の相互動作。	*
単一アソシエーションオブジェクト (S A O) : 単一のアプリケーションアソシエーション を通じて相手と通信するために必要となる機能。	* *
単一アソシエーション制御機能 (S A C F) : 単一のアプリケーションアソシエーションを通じて相手と通信するために用いられる A S E の利用を制御する規則と規制。	* *
ディレクトリアクセスプロトコル (D A P) : T T C 標準 J T - X 5 0 0 (1 2 章) を参照。	*
ディレクトリシステムプロトコル (D S P) : T T C 標準 J T - X 5 0 0 (1 2 章) を参照。	*
ディレクトリ情報木 (D I T) : T T C 標準 J T - X 5 0 0 (1 4 章) を参照。	*

ディレクトリユーザエージェント (DUA) : TTC 標準 JT - X 5 0 0 (4 . 2 章) を参照。	*
電気通信管理ネットワーク (TMN) : 電気通信ネットワークの管理機能に関連する情報の転送及び処理のための手段を提供するエンティティ。	* *
特殊リソース機能 (SRF) : サービスを提供するために使用するリソースの制御とアクセスを提供する機能のセット。	* *
トランザクション : TC 提供者間のアソシエーション。	*
トランザクション機能応用部 (TC) : 通信網での自動オペレーションをサポートする同位間プロトコル。	*
ネームフォーム : TTC 標準 JT - X 5 0 0 の「名前形式」を参照。	*
複数アソシエーション制御機能 (MACF) : アプリケーションエンティティインボケーション(AEI)内で、同位間通信の調整を管理する規則を示す。	* *
物理エンティティ (PE) : 各機能単位の物理的なエンティティ	*
物理プレーン : 機能エンティティを実現するエレメントとそれらのインタフェースを含む統一された機能。	*
分散機能プレーン (DFP) : 機能エンティティとそれらの関連を含む統一された機能。	*
網 : ITU - T Vol . I 分冊 I.3 参照。	*
有限状態機構 (FSM) : 有限数な状態と状態間の特定の遷移をもつシステム。	*
ISDNユーザパート (ISDN - UP) : サービス統合デジタル網での音声および非音声アプリケーションのための基本ベアラサービスや付加サービスのサポートをするために要求される信号化機能を提供する No . 7 信号方式プロトコル。	* * *
SCF 呼状態モデル (SCSM) : SCF における呼処理の状態モデル。	*
SCF 管理エンティティ (SCME) : SCME 制御と複数の SCME FSM インスタンスから構成され、SCF から受信されたオペレーションの実行の管理を制御するエンティティの機能。	* *

2 . 略語

この標準の目的のために、次の略語が適用される。

AC:	Application Context	*
ACI:	Access Control Information	*
ACN:	Application Context Negotiation	*
AD:	Adjunct	*
AE:	Application Entity	*
AEI:	Application Entity Invocation	*
APDU:	Application Protocol Data Unit	*
ASE:	Application Service Element	*
CCAF:	Call Control Agent Function	*
CCF:	Call Control Function	*
CS-2:	Capability Set-2	*
CUSF:	Call Unrelated Service Function	*
DAP:	Directory Access Protocol	*
DFP:	Distributed Functional Plane	*
DHA:	Dialogue Handler	*
DISP:	Directory Information Shadowing Protocol	*
DIT:	Directory Information Tree	*
DPC:	Destination Point Code	*
DSA:	Directory System Agent	*
DSP:	Directory System Protocol	*
DSS1:	Digital Subscriber Signaling No. 1 Protocol	*
DUA:	Directory User Agent	*
FE:	Functional Entity	*
FEAM:	Functional Entity Access Manager	*
FSM:	Finite State Machine	*
GT:	Global Title	*
GTAI:	Global Title Address Information	*
IN:	Intelligent Network	*
INAP:	Intelligent Network Application Protocol	*
ISDN:	Integrated Services Digital Network	*
ISDN-UP:	ISDN User Part	*
ISO:	International Organization for Standardization	*
ISPT:	Inter-network Service Profile Transfer	*
ITU-T:	International Telecommunication Union - Telecommunication Standardization	*
MACF:	Multiple Association Control Function	*
MTP:	Message Transfer Part	*
NSSR:	Non-Specific Subordinate References	*
OSI:	Open Systems Interconnection	*
PDU:	Protocol Data Unit	*
PE:	Physical Entity	*
PSTN:	Public Switched Telephony Network	*

RDN:	Relative Distinguished Name	*
ROS:	Remote Operations	*
ROSE:	Remote Operations Service Element	*
SACF:	Single Association Control Function	*
SAO:	Single Association Object	*
SAP:	Service Access Point	*
SCCP:	Signaling Connection Control Part	*
SCF:	Service Control Function	*
SCP:	Service Control Point	*
SCSM:	Service Control Function Call State Model	*
SCSM-SDF:	SDF Related SCSM	*
SCME:	SCF Management Entity	*
SDF:	Service Data Function	*
SDP:	Service Data Point	*
SDSM:	Service Data Function Call State Model	*
SDSM-ShSSi:	SDSM Shadowing Supplier initiated Supplier	*
SDSM-ShSCi:	SDSM Shadowing Supplier initiated Consumer	*
SDSM-ShCSi:	SDSM Shadowing Consumer initiated Supplier	*
SDSM-ShCCi:	SDSM Shadowing Consumer initiated Consumer	*
SLEE:	Service Logic Execution Environment	*
SLP:	Service Logic Processing Program	*
SLPI:	Service Logic Processing Program Instance	*
SN:	Service Node	*
SRF:	Specialized Resource Function	*
SS7:	Signaling System No.7	*
SSCP:	Service Switching and Control Point	*
SSN:	Sub System Number	*
SSF:	Service Switching Function	*
SSP:	Service Switching Point	*
TC:	Transaction Capabilities	*
TMN:	Telecommunications Management Network	*

付録Ⅱ データモデリング

(TTC標準JT-Q1228に対する)

Ⅱ.1 はじめに

Ⅱ.1.1 目的と範囲

この付録では、Inter-Network Service Profile Transfer (ISPT) において利用可能であるオブジェクトモデル化の手法を示す。

Ⅱ.1.2 前提

Ⅱ.1.2.1 移動体通信サービスと加入者識別子

Ⅱ.1.2.1.1 移動体通信加入者固有識別子

同じ移動体通信サービス提供者に属する個々の加入者を識別するため、移動体通信サービス加入者は、数値表現される固有の識別子を持たなければならない。この加入者識別子は、ローカル(一つのサービス提供者)に意味を持つ。つまり個々のサービス提供者は、他のサービス提供者において付与される固有の加入者識別子を知る必要はない。

識別子の最大長については互いのサービス提供者間で合意すべきであるが、その最大値を超えない範囲で付けられる加入者識別子の長さは、ローカルな決め事である。

Ⅱ.1.2.1.2 移動体通信サービス提供者プリフィックス

個々の移動体通信サービス提供者は、固有の数値識別子(プリフィックス)を持たなければならない。サービス提供者のプリフィックスは、グローバル(協力関係にある複数のサービス提供者間)に意味を持つ。つまり協力関係にある個々の移動体通信サービス提供者は、互いに他のサービス提供者のプリフィックスを理解しなければならない。

プリフィックスの最大長の設定については互いに合意すべきである。

個々のローカル加入者の固有識別子につけるプリフィックスとして、そのサービス提供者の識別子を用いることにより、全ての移動体通信サービス加入者は、固有かつ明確に識別できる。

Ⅱ.1.2.1.3 移動体通信サービス提供者サブ-プリフィックス

ある移動体通信サービス提供者が、複数のDSAを用いて加入者のサービスデータを管理する例のように、加入者を論理的に分類可能とする必要がある場合には、サービス提供者サブ-プリフィックスを用いることができる。サービス提供者サブ-プリフィックスは、ローカル(一つのサービス提供者)に意味を持つ。つまり、協力関係にある移動体通信サービス提供者間では、互いのサービス提供者サブ-プリフィックスを理解する必要はない。

識別子の最大長については互いのサービス提供者間で合意すべきであるが、その最大値を超えない範囲で付けられる加入者識別子の長さは、ローカルな決め事である。

付加的なサブ-プリフィックスの利用方法は、本付録に定義されていない。この利用方法が将来的に必要であると想定しているわけではないが、これを定義し、実装することは困難なことではない。

II.1.2.1.4 符号化

ローミングユーザの固有識別子、サービス提供者プリフィックスおよびサブ-プリフィックスを在圏網に知らせる手段は、本付録の範囲外である。本付録は、ユーザの固有識別子や、サービス提供者プリフィックス、サブ-プリフィックスを個々に識別可能とするために、オブジェクト識別子、長さおよび値のようないくつかの符号化タイプについて言及することを目的とする。

II.1.2.2 プロファイルエントリの開示

あるサービス提供者に属する加入者エントリの全リストが、その他のサービス提供者にて利用可能とするように要求することは、望ましいものではない。ローミングしているユーザのプロファイルのみ、開示すべきである。

II.2 D I Tスキーマ

II.2.1 T T C 標準 J T - X 5 0 0 D I T

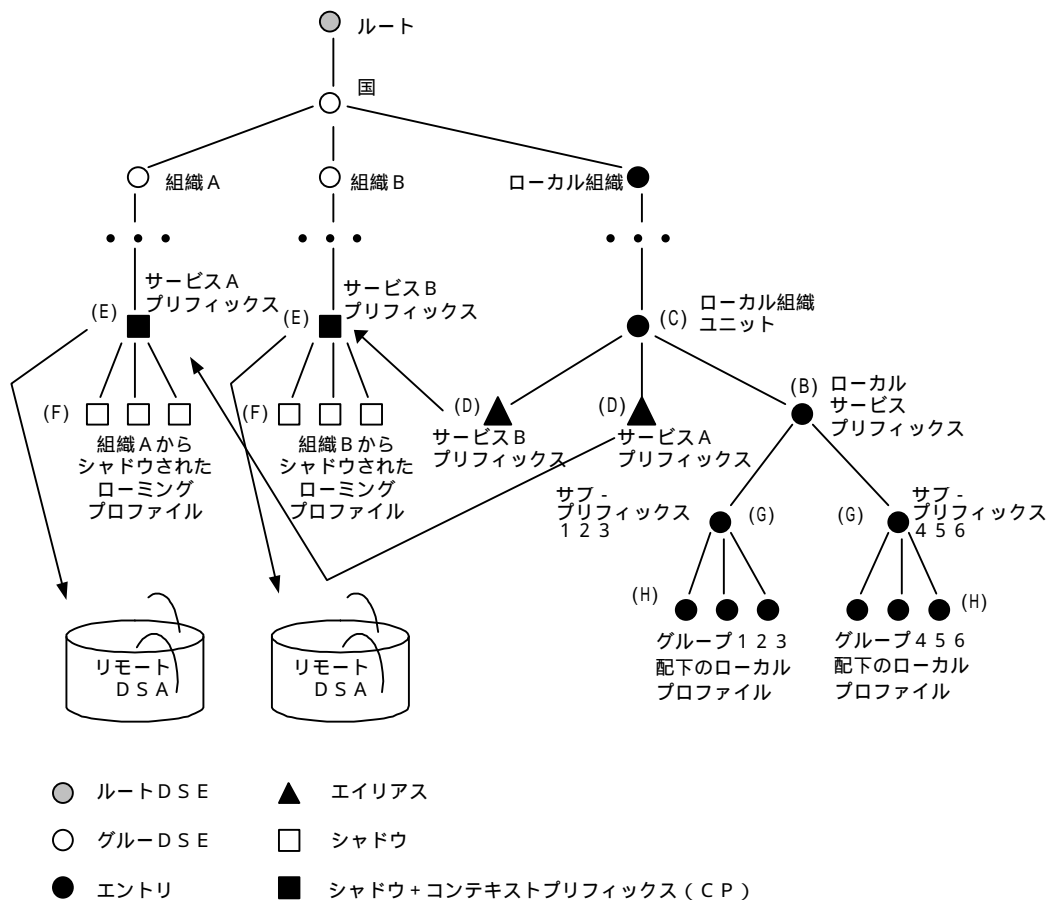
付図 II - 1 / J T - Q 1 2 2 8 に示す T T C 標準 J T - X 5 0 0 D I T は本付録で述べる D I T スキーマを示す。

II.2.1.1 ローカルプロファイルのロケーション

あるサービス提供者のプリフィックスは、サービス提供者のローカル D I T において、配下に全てのローカル加入者プロファイルを格納するエントリノード（付図 II - 1 / J T - Q 1 2 2 8 にて(B)とラベルされたノードを参照）を命名するために用いられる。サービス提供者プリフィックスエントリは、値が `inMobilityServicePrefix` であるオブジェクトクラスを含む。

ユーザプロファイルは、更にサービス提供者のサブ-プリフィックスにより分類することができる。仮にサブ-プリフィックスを使用する場合は、付加的な Relative Distinguished Name (R D N) と、ユーザの固有識別子とサービス提供者のプリフィックス間の D I T におけるノード ((G) とラベルされたノード) により実現される。サービス提供者サブ-プリフィックスエントリが使用されている場合、値が `inMobilityServiceSubPrefix` であるオブジェクトクラスを含む。サブ-プリフィックスエントリはサービス提供者サブ-プリフィックス識別子を用いて命名される。

ローカルプロファイルエントリ ((H) とラベルされたノード) は、値が `inMobilityUserProfile` であるオブジェクトクラスを含む。ローカル加入者エントリは、ローカル加入者の固有移動体通信サービス識別子を用いて命名される。



付図 II - 1 / J T - Q 1 2 2 8
 (I T U - T Q . 1 2 2 8)
 移動体通信サービス提供者用ローカル D I T の例

II.2.1.2 リモート加入者プロファイルのロケーション

あるサービス提供者の加入者プロファイルのロケーションは、その他の協力関係にあるサービス提供者に利用可能なものでなければならない。そこで、リモート D S A において格納されるプロファイルの上位エントリ (E) とラベルされたノード) は、ローカル D S A 上にシャドウされなければならない。このシャドウ D S E エントリが不完全な下位知識を持つ場合に限り、在圏網は、continuation reference をこのエントリ内で発見される supplierKnowledge 属性から生成することができる。この continuation reference は、実際のプロファイルが存在するローミングユーザのホーム D S A を指し示す。

II.2.1.3 ローミング時の加入者プロファイルのロケーション

ローミング加入者 (locally visiting subscriber) は、前述のノード (E) とラベルされたノード) の配下にシャドウされる、加入者プロファイルエントリ (F) とラベルされたノード) を持つ。個々のプロファイルエントリは、加入者が属するホームのサービス提供者にそれぞれ対応している。

II.2.1.4 メッセージフローの削減

ユーザが別の網にローミングする場合、在圏網は、ローミングしているユーザのホーム網におけるユーザプロファイルを更新しなければならない。これにより、ユーザのプロファイルが在圏網に転送される。

ユーザのプロファイルを更新するために、在圏網はローミングしているユーザのDNを確定しなければならない。メッセージフローを削減するために、DAPにおける複数の search オペレーションを用いることなく、このDNは解決できる。加入者識別子とディレクトリ番号のDNへのマッピングは、固定長番号計画の使用によって実現できる。

前節の前提により、ローミングしているユーザの固有識別子とプリフィックス（必要であればサブ-プリフィックスも）は、在圏網において認識できる。これらの識別子はローミングしているユーザのDNを構成するRDNとして用いることができる。仮のDNは、ホーム網のプリフィックスエントリ（ノード(B)）に対する上位ノードのDNに、これらのRDNを追加することにより生成可能である。

協力関係にあるそれぞれのサービス提供者に対するエイリアスエントリ（(D)とラベルされたノード）が、ローカルサービスに関する加入者のプロファイルを配下に格納するノード（ノード(B)）に隣接して格納されているとする。また、これらのエイリアスエントリは、サービス提供者プリフィックス番号を用いて命名され、ローカルDIT中の他の場所に格納されたそれに対応するノード（(E)とラベルされたノード）を指し示すために用いられると仮定する。

DAPオペレーションで使用される場合、ローミングユーザの仮のDNは、ローミングユーザのホーム網プリフィックスのシャドウであるDITにおけるノードを指し示すために展開される。このシャドウDSEエントリが不完全な下位知識を持つ場合に限り、在圏網は、continuation reference をこのエントリ内で発見される supplierKnowledge 属性から生成することができる。continuation reference は、実際のプロファイルが存在するローミングユーザのホームDSAを指し示すものである。この操作はローミングユーザのホームDSAにその後 chain されるものである。

II.2.2 オブジェクトクラス

II.2.2.1 inMobilityUserProfile

inMobilityUserProfile オブジェクトクラスは、プロファイル情報を格納するために定義されている。以下のASN.1定義は、inMobilityUserProfile オブジェクトクラスを記述するために使用される。

```
inMobilityUserProfile OBJECT-CLASS ::= {
    SUBCLASS OF { top}
    MUST CONTAIN { inMobilityID |
                  inMobilityPIN |
                  <other mandatory attributes>}
    MAY CONTAIN { <optional attributes>}
    ID          Id-oc-inMobilityUserProfile}
```

inMobilityID 属性は、識別属性である。

このタイプのエントリは、inMobilityServiceProvider タイプ又は inMobilitySubscriberGroup タイプのエントリ配下のみに配置される。

II.2.2.2 inMobilityServiceProvider

このオブジェクトクラスは、配下に inMobilityUserProfile タイプと inMobilitySubscriberGroup タイプのオブジェクトを格納できるノードを定義するために作成されている。以下の A S N . 1 定義は、inMobilityServiceProvider オブジェクトクラスを記述している。

```
inMobilityServiceProvider    OBJECT-CLASS ::= {
    SUBCLASS OF                { top}
    MUST CONTAIN               { inMobilityPrefix | <other mandatory attributes>}
    MAY CONTAIN                { <optional attributes>}
    ID                         Id-oc-inMobilityServiceProvider}
```

inMobilityPrefix 属性は、識別属性である。

このタイプのエントリは、リモート D S A に格納されているプロファイルエントリ又はサブ-プリフィックス（使用されている場合）を指し示している、non-specific subordinate reference（N S S R）を含んでもよい。

このオブジェクトクラス配下のエントリは、inMobilityUserProfile タイプ又は inMobilitySubscriberGroup タイプでなければならない。

II.2.2.3 inMobilitySubscriberGroup

このオブジェクトクラスは、配下に inMobilityUserProfile タイプのオブジェクトを格納できるノードを定義するために作成された。以下の A S N . 1 定義は、inMobilitySubscriberGroup オブジェクトクラスを記述する。

```
inMobilitySubscriberGroup    OBJECT-CLASS ::= {
    SUBCLASS OF                { top}
    MUST CONTAIN               { inMobilitySubPrefix | <other mandatory attributes>}
    MAY CONTAIN                { <optional attributes>}
    ID                         Id-oc-inMobilitySubscriberGroup}
```

inMobilitySubPrefix 属性は、識別属性である。

このオブジェクトクラス配下のエントリは、inMobilityUserProfile タイプでなければならない。

II.2.3 属性タイプ

II.2.3.1 inMobilityID

この属性は、ある特定のサービス提供者又はサービス加入者グループの移動体通信ユーザを個々に識別するために使用される。

inMobilityID のための A S N . 1 定義を以下に示す。

```
inMobilityID                 ATTRIBUTE ::= {
    WITH SYNTAX                Digits (SIZE(lb-inMobilityID..ub-inMobilityID))
    EQUALITY MATCHING RULE    octetStringMatch
    ID                         id-at-inMobilityID}
```

II.2.3.2 inMobilityPIN

この属性は、移動体通信ユーザの P I N 番号を格納するために使用される。

inMobilityPIN WITH SYNTAX ID	ATTRIBUTE ::= { userPassword (SIZE lb-inMobilityPIN..ub-inMobilityPIN) id-at-inMobilityPIN}
------------------------------------	---

II.2.3.3 inMobilityPrefix

この属性により、移動体通信サービス提供者を個々に識別する。

inMobilityPrefix WITH SYNTAX EQUALITY MATCHING RULE ID	ATTRIBUTE ::= { Digits (SIZE(lb-inMobilityPrefix.. ub-inMobilityPrefix)) octetStringMatch id-at-inMobilityPrefix}
---	---

II.2.3.4 inMobilitySubPrefix

この属性は、移動体通信加入者を分類するために使用される。

inMobilitySubPrefix WITH SYNTAX EQUALITY MATCHING RULE ID	ATTRIBUTE ::= { Digits (SIZE(lb-inMobilitySubPrefix.. ub-inMobilitySubPrefix)) octetStringMatch id-at-inMobilitySubPrefix}
--	--

II.2.4 D I T 構造定義

II.2.4.1 ネームフォーム

ネームフォームは、特定のオブジェクトクラスの R D N として使用される属性を規定する。

II.2.4.1.1 inMobilityUserProfileNameForm

以下のネームフォーム定義は inMobilityID が inMobilityUserProfile オブジェクトクラスについての許容された識別属性であることを示す。

inMobilityUserProfileNameForm WITH ATTRIBUTES ID	NAME-FORM ::= { inMobilityID id-nf-inMobilityUserProfileNameForm}
--	---

II.2.4.1.2 inMobilityServiceProviderNameForm

以下のネームフォーム定義は、 inMobilityPrefix が inMobilityServiceProvider オブジェクトクラスについての許容された識別属性であることを示す。

inMobilityServiceProviderNameForm NAMES WITH ATTRIBUTES ID	NAME-FORM ::= { inMobilityServiceProvider inMobilityPrefix id-nf-inMobilityServiceProviderNameForm}
---	--

II.2.4.1.3 inMobilitySubscriberGroupNameForm

以下のネームフォーム定義は、inMobilitySubPrefix が inMobilitySubscriberGroup オブジェクトクラスについての許容された識別属性であることを示す。

```
inMobilitySubscriberGroupNameForm NAME-FORM ::= {
    NAMES                               inMobilitySubscriberGroup
    WITH ATTRIBUTES                     inMobilitySubPrefix
    ID                                   id-nf-inMobilitySubscriberGroupNameForm}
```

II.2.4.2 構造規則

構造規則は D I T 中での許容された下位及び上位のエントリを規定する規則である。付図 II - 2 / J T - Q 1 2 2 8 で書かれている次に示す構造規則は、移動体通信サービスに要求される構造規則を定義する基本として使用され得る。

```
sr1      STRUCTURE-RULE ::= {
          NAME-FORM          countryNameForm
          ID                  1}

sr2      STRUCTURE-RULE ::= {
          NAME-FORM          orgNameForm
          SUPERIOR RULES    sr1
          ID                  1}

...

sr<n>    STRUCTURE-RULE ::= {
          NAME-FORM          inMobilityServiceProviderNameForm
          SUPERIOR RULES    sr<n-1>
          ID                  <n>}

sr<n+1>  STRUCTURE-RULE ::= {
          NAME-FORM          inMobilitySubscriberGroupNameForm
          SUPERIOR RULES    sr<n>
          ID                  <n+1>}

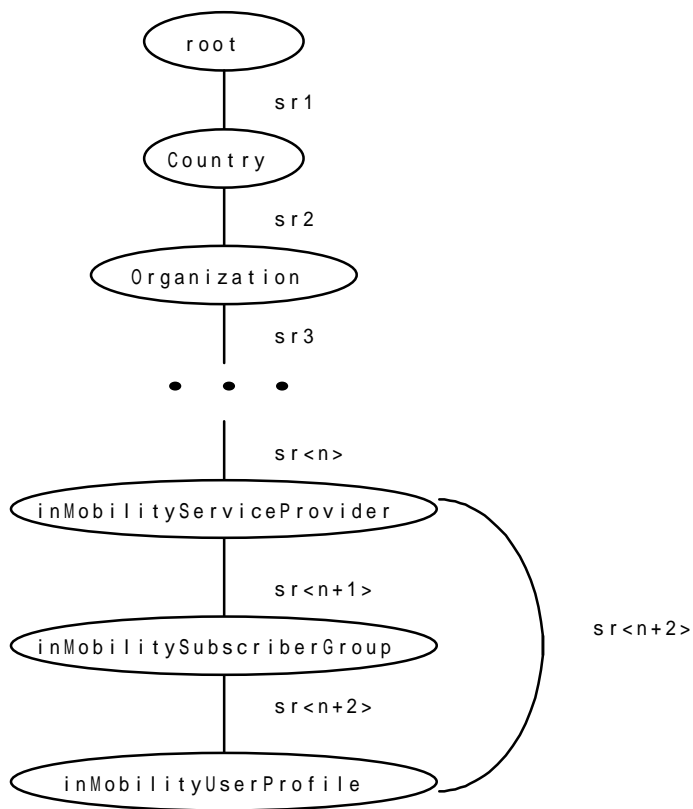
sr<n+2>  STRUCTURE-RULE ::= {
          NAME-FORM          inMobilityUserProfileNameForm
          SUPERIOR RULES    sr<n>, sr<n+1>
          ID                  <n+2>}
```

II.2.4.3 オブジェクト識別子の割当て

次のオブジェクト識別子割当てが T T C 標準 J T - X 5 0 0 の移動体通信オブジェクトを特定する起点として使用され得る。

```
id-at-inMobilityID          OBJECT IDENTIFIER ::= {id-at-inMobility 0}
id-at-inMobilityPIN         OBJECT IDENTIFIER ::= {id-at-inMobility 1}
id-at-inMobilityPrefix     OBJECT IDENTIFIER ::= {id-at-inMobility 2}
id-at-inMobilitySubPrefix  OBJECT IDENTIFIER ::= {id-at-inMobility 3}
id-oc-inMobilityUserProfile OBJECT IDENTIFIER ::= {id-oc-inMobility 0}
id-oc-inMobilityServiceProvider OBJECT IDENTIFIER ::= {id-oc-inMobility 1}
```

id-oc-inMobilitySubscriberGroup	OBJECT IDENTIFIER ::= {id-oc-inMobility 2}
id-nf-inMobilityUserProfileNameForm	OBJECT IDENTIFIER ::= {id-nf-inMobility 0}
id-nf-inMobilityServiceProviderNameForm	OBJECT IDENTIFIER ::= {id-nf-inMobility 1}
id-nf-inMobilitySubscriberGroupNameForm	OBJECT IDENTIFIER ::= {id-nf-inMobility 2}



付図Ⅱ- 2 / JT - Q 1 2 2 8
 (ITU - T Q . 1 2 2 8)
 構造規則

付録Ⅲ SPKMアルゴリズムの例

#

第1版作成協力者(1999年1月27日現在)

(敬称略)

第一部門委員会

部門委員長	菅 俊直	K D D (株)
副部門委員長	和泉 俊勝	日本電信電話(株)
副部門委員長	林 和行	(株)日立製作所
委員	竹原 啓五	第二電電(株)
委員	小林 昌宏	東京通信ネットワーク(株)
委員	山口 健二	日本電気(株)
委員	坪井 洋治	WG1-1委員長・富士通(株)
委員	片野 俊樹	WG1-1副委員長・日本電信電話(株)
委員	大塚 宗丈	WG1-2委員長・日本電信電話(株)
委員	池田 一雄	WG1-2副委員長・沖電気工業(株)
委員	平野 郁也	WG1-2副委員長・日本無線(株)
委員	高瀬 晶彦	WG1-4委員長・(株)日立製作所
委員	奈須野 裕	WG1-4副委員長・日本テレコム(株)
委員	吉村 勝仙	WG1-4副委員長・日本電信電話(株)
委員	大宮 知己	WG1-IN委員長・日本電信電話(株)
委員	吉田 龍彦	WG1-TMN委員長・日本電信電話(株)
委員	益田 淳	WG1-TMN副委員長・K D D (株)
委員	鈴木 茂房	WG1-UPT委員長・日本電信電話(株)

(敬称略)

第一部門委員会 第一専門委員会

専門委員長	坪井 洋治	富士通(株)
副専門委員長	片野 俊樹	日本電信電話(株)
委員	泊 哲郎	国際デジタル通信(株)
委員	平海 孝志	第二電電(株)
委員	佐口 雅広	東京通信ネットワーク(株)
委員	奈須野 裕	日本テレコム(株)
委員	大羽 巧	日本電信電話(株)
委員	平田 昇一	N T T 移動通信網(株)
委員	黄木 寛之	(株)東京デジタルホン
委員	野村 忠勝	中部テレコミュニケーション(株)
委員	鈴木 和美	(株)ツーカーセルラー東京
委員	矢沢 勝彦	(株)ツーカーホン関西
委員	紺谷 武夫	日本移動通信(株)
委員	片貝 義人	東京テレメッセージ(株)
委員	目黒 喜治	大阪メディアポート(株)
委員	近 義起	D D I 東京ポケット電話(株)
委員	東海林 尚美	(株)アステル東京
委員	山田 博	(株)インテック
委員	後藤 雅徳	沖電気工業(株)
委員	田村 慶章	(株)東芝
委員	長谷川 茂夫	日本ルーセント・テクノロジー(株)
委員	山口 健二	日本電気(株)
委員	境 穰	日本無線(株)
委員	高木 淳	ノーザンテレコムジャパン(株)
委員	新保 勲	(株)日立製作所
委員	内田 陽子	富士通(株)
委員	大塚 晃	三菱電機(株)
委員	住田 正臣	日本エリクソン(株)
委員	浜田 啓嗣	日本情報通信コンサルティング(株)
委員	竹原 啓五	第二電電(株)
委員	横田 孝弘	K D D (株)
委員	松本 弘行	K D D (株)
委員	江崎 修司	日本電信電話(株)
委員	幕田 和彦	K D D (株)
事務局	加藤 敏郎	T T C 事務局

J T - Q 1 2 2 8 検討グループ (S W G 5)

リーダー	江崎 修司	日本電信電話 (株)
特別専門委員	篠原 健之	K D D (株)
特別専門委員	吉岡 一明	K D D (株)
特別専門委員	中村 友久	第二電電 (株)
委員	増田 知範	東京通信ネットワーク (株)
特別専門委員	柳下 健二	日本テレコム (株)
委員	澤田 政宏	N T T 移動通信網 (株)
特別専門委員	市原 直彦	(株) 東京デジタルホン
委員	野村 忠勝	中部テレコミュニケーション (株)
特別専門委員	石田 真明	(株) ツーカーホン関西
特別専門委員	小林 靖明	日本移動通信 (株)
特別専門委員	渡辺 幸雄	D D I 東京ポケット電話 (株)
特別専門委員	仲川 由子	沖電気工業 (株)
特別専門委員	松浦 豊	日本電気 (株)
特別専門委員	柴 博昭	日本電気 (株)
委員	境 穰	日本無線 (株)
特別専門委員	佐久間 哲雄	(株) 日立製作所
委員	内田 陽子	富士通 (株)
特別専門委員	本持 智子	富士通 (株)
特別専門委員	木下 裕介	三菱電機 (株)
委員	住田 正臣	日本エリクソン (株)