

OSIセッションレーヤ標準のインプリメント

鈴木健二 加藤聰彦
国際電信電話株式会社 研究所

1. まえがき

ISO、CCITTでは、異機種コンピュータ・端末間通信を実現するために、開放型システム間相互接続(OSI: Open Systems Interconnection)の標準化を進めており、セッション層の標準化もほぼ確定している。現在、セッション層のサービス定義とプロトコル仕様は、各々、CCITTでは勧告X.215, X.225、ISOでは、DIS(Draft International Standard) 8326, 8327となっている^{[1][2]}。

筆者等は、公衆パケット網においてMHS(Message Handling System)やディレクトリサービス等の高度通信処理サービスを提供する網内HOSTの構築過程で、先に実装済みのOSIトランスポートプロトコルのクラス0、1、2、3プログラム^[3]とインタフェースして動作するセッションプロトコルを実装した^[4]ので、以下にその概要と問題点について報告する。

2. セッション層の役割と機能

図1にOSI参照モデルを示す。セッション層(SL)は、

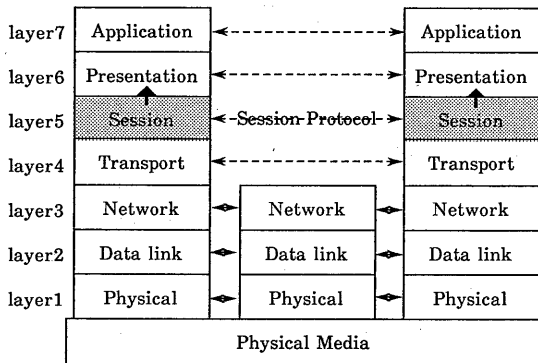


図1 OSI参照モデルにおけるセッションレーヤの位置付け(図中↑はサービスを示す)

下位から第5層に位置し、業務内容に応じて多種多様に存在するアプリケーションプロセス(以後プロセスと呼ぶ)間でプロセス間の同期をとりながら、プロセスにとって何らかの意味をもつデータ(情報)の送受方法を管理する。これは、一般的に対話(ダイア

ログ: dialogue)の管理と、総称されることもある。具体的には以下のような機能を持つ

- ① 相手SSユーザ(セッション層がその上位層に提供するサービス要素を使用する実体をモデル化したもの)とのセッションコネクション(SC)を確立し、データ転送、SCの解放を管理する。
- ② データ転送、同期、SCの解放を行うに必要な送信権(token)の折衝や管理、および半二重や全二重データ転送の設定を行う。
- ③ 一連のデータ転送の流れの中に、同期点を挿入し、エラーが生じた場合には、双方で合意した同期点から再開する機能。
- ④ 一連のデータ転送を途中でやめ、後に再開する機能

ここでセッション層の機能を更に明確にするため、トークン、同期とダイアログ単位、アクティビティの概念について説明する。

(1) トークン

トークンはSCの属性で、SSユーザが特定のセッションサービスを起動する権利を示し、一時点には一方のSSユーザにだけ割り当てられる。このトークンはSSユーザからの要求に応じてトークン管理のプロトコル要素により双方のセッションエンティティ間でやりとりされ、その所在がSSユーザに通知される。トークンにはデータ、同期マイナー、メジャー/アクティビティ、リリースの4種類がある。

(2) 同期とダイアログ単位

SSユーザは、送信する一連のデータの中に同期点を挿入でき、各同期点は、SSプロバイダ(SSユーザに対応する用語でセッションサービスの提供者をモデル化したもの)が管理する番号で識別する。この同期点の意味付けは、SSユーザ間の通信毎に自由に定められる。この同期点には、大同期点と小同期点の2種類がある。大同期点は、一連のデータの流れをいく

つかのダイアログ単位に区切るもので、一つの大同期点はダイアログ単位の終了と次のダイアログ単位の開始を意味する。この大同期点には必ず送達確認があるため、例えば一つのドキュメント転送の終りというように、通常大同期点の前後で通信の意味付けをかえられる。

一方、小同期点は、ダイアログ単位の中でデータ転送を構造化するために用いられる。

(3) アクティビティ

アクティビティ(Activity)もプロセス間で同期を保ちながら行うデータ転送の構造化をさらに進めるために導入された概念である。これは、プロセス間通信で行う一つの仕事の部分をグループ化して捉える概念ともいうことができ、一つのアクティビティは複数のダイアログ単位からなる。一つのSC上には同時には唯一のアクティビティしか存在できないが、このアクティビティは複数のSCにまたがることことができる。また、アクティビティは中断でき、かつ同一ないしは後続するSCで再開できる。

以上のように、セッション層では、各種プロセス間通信の秩序ある対話を管理するため、データ転送の機能を構造化し、SSユーザが独自の意味付けができる機構を提供する。ダイアログ単位、アクティビティおよび同期の関係を図2に示す。一例をドキュ

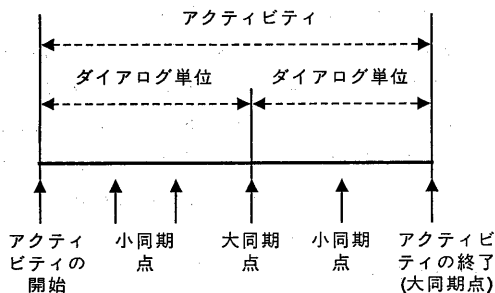


図2 アクティビティとダイアログ単位の関係

メント転送にとる。ある場合には、章ごとに小同期点を対応させ、ドキュメントの終了を大同期点に対応させる。一方、ある場合には頁を小同期点に、章を大同期点、そして、ドキュメント全体をアクティビティに対応させるかもしれない。プロセス間で何

か意味ある区切りをつけるためにこれらダイアログ単位、アクティビティが使い分けられる。

3. セッションレイヤ標準の概要

3.1 機能単位

セッションプロトコルでは、そのプロトコル要素となるSPDU(Session Protocol Data Unit)のうち、相互に関連するものを機能単位(FU: Functional Unit)としてグループ化している。ここで各FUについて概説する。

(1) カーネルFU

SCの確立、データ転送、SCの解放に必要な基本的なプロトコル要素の集合。本FUは、どのSCでも必要とされる。

(2) ネゴシエートリリースFU

SCの解放指示を受信した際、これを拒否し、SCの継続的使用を許すための機能。本FUを使用する場合にはリリーストークン(release token)が有効でなければならない。

(3) 半二重FU

半二重通信モードの機能を提供する。データトークン(data token)が関連づけられており、データトークンを有する側が普通データの送信の権利をもつ。

(4) 全二重FU

データトークンには制御されず、普通データの両方向同時転送を行う。

(5) 優先データFU

普通データの流れとは別にデータを転送するもので、優先の意味は、後続する普通データより必ず先に相手に到達することをいう。トランスポート優先データが使用可能な場合に用いることができる。

(6) タイプトデータFU

SSユーザがデータトークンの有無にかかわらずデータを送信できる機能を提供する。

(7) ケーパビリティデータ交換FU

アクティビティ管理FUが用いられ、かつアクティビティが進行中でない場合に使用される。本FUは、受信確認を必要とするデータ転送の機能を提供し、例えば相手の受信端末能力の折衝に用いられる。

(8) 小同期FU

一連の普通データ転送の途中で、SSユーザが小同期点を挿入する機能を提供する。小同期点は同期点順序番号で識別される。同期マイナートークン(synchronize-minor token)が関連づけられ、このトークンの所有者のみが小同期点を送出できる。小同期点は必ずしも送達確認を必要としない。

(9) 大同期FU

一連の普通データ転送の途中で、SSユーザが大同期点を挿入する機能を提供する。大同期点も同期点順序番号により識別される。大同期点には必ず送達確認を必要とし、送達確認が得られるまで、新たな普通データ、優先データの送出ができないため、大同期点の前後でデータの流れが明確に区別される。メジャー/アクティビティトークン(major/activity token)が関連づけられ、本トークンの所有者のみが大同期点を設定できる。

(10) 再同期FU

SSユーザが同期点の再設定およびトークンの再配置を行う機能を提供する。現在のダイアログ単位を異常終了させるため、データの紛失を伴うことがある。従って再同期には、再開する同期点の設定方法によりアバンダン(abandon)、リスタート(restart)、セット(set)の3種類のオプションがある。

(11) 例外報告FU

SSプロバイダないしはSSユーザが検出したエラーを、各々、SSユーザおよび相手SSユーザに通知する機能を提供する。本FUは半二重FUが使用される場合にのみ使用できる。

(12) アクティビティ管理FU

プロセスが行う仕事の一部分の構成単位を管理する機能を提供する。メジャー/アクティビティトークンと関連があり、本トークンの所有者はアクティビティを開始、終了、再開、中断、廃棄することができる。

各機能単位とSPDUの関係を表1に示す。

これらの機能単位の使用は、SC確立時に双方のSSユーザ間で折衝して、自由に組み合わせることができるが、

- ① カーネルは必ず使用する

表1 セッションプロトコルの機能単位とSPDU

Functional unit	SPDU code	SPDU name
Kernel	CN	CONNECT
	AC	ACCEPT
	RF	REFUSE
	FN	FINISH
	DN	DISCONNECT
	AB	ABORT
	AA	ABORT ACCEPT
	DT	DATA TRANSFER
	PT	PLEASE TOKENS
	GI	GIVE TOKENS
	Half-duplex	GI
Duplex Exceptions	ER	No additional associated SPDU EXCEPTION REPORT
	ED	EXCEPTION DATA
Typed data	TD	TYPED DATA
Negotiated release	NF	NOT FINISHED
	PT	PLEASE TOKENS
	GI	GIVE TOKEN
Minor synchronize	MIP	MINOR SYNC POINT
	MIA	MINOR SYNC ACK
	PT	PLEASE TOKENS
Major synchronize	GI	GIVE TOKENS
	MAP	MAJOR SYNC POINT
	MAA	MAJOR SYNC ACK
	PR	PREPARE
Resynchronize	PT	PLEASE TOKENS
	GI	GIVE TOKENS
	RS	RESYNCHRONIZE
	RA	RESYNCHRONIZE ACK
	PR	PREPARE
Expedited Activity management	EX	EXPEDITED DATA
	AS	ACTIVITY START
	AR	ACTIVITY RESUME
	AI	ACTIVITY INTERRUPT
	AIA	ACTIVITY INTERRUPT ACK
	AD	ACTIVITY DISCARD
	ADA	ACTIVITY DISCARD ACK
	AE	ACTIVITY END
	AEA	ACTIVITY END ACK
	PR	PREPARE
	PT	PLEASE TOKENS
	GI	GIVE TOKENS
	GTC	GIVE TOKENS CONFIRM
	GTA	GIVE TOKENS ACK
	Capability data exchange	CD
CDA		CAPABILITY DATA ACK

- ② ケーパビリティデータ交換FUを使用する場合は、アクティビティ管理FUも使用する。
 - ③ 例外報告FUを必要とする場合は、半二重FUも必要である。
- という条件がある。

3.2 セッションサービス

表2に、セッションサービス定義で規定されたサービスプリミティブを示す。セッションサービス定義では、S-CONNECTプリミティブのSession Requirementパラメータで前述した機能単位の使用を双方のSSユーザ間で折衝する。このため、SSユーザがセッションサービスを使用する際に、ガイドラインを与えるため、あらかじめ想定される使用方法として3種類のサービスサブセットを規定している。

表2 セッションサービス定義のプリミティブ

SERVICE	PRIMITIVES	PARAMETERS
Session Connection	S-CONNECT req S-CONNECT ind S-CONNECT resp S-CONNECT conf	Session connection identifier, Calling / Called SSAP identifiers, Result, QOS, Session requirements, Sync-point serial number, Initial token settings, SS-user data
Normal Data Transfer	S-DATA req S-DATA ind	SS-user data
Expedited Data Transfer	S-EXPEDITED-DATA req S-EXPEDITED-DATA ind	SS-user data
Typed Data Transfer	S-TYPED-DATA req S-TYPED-DATA ind	SS-user data
Capability Data Exchange	S-CAPABILITY-DATA req S-CAPABILITY-DATA ind S-CAPABILITY-DATA resp S-CAPABILITY-DATA conf	SS-user data
Give Tokens	S-TOKEN-GIVE req S-TOKEN-GIVE ind	Tokens
Please Tokens	S-TOKEN-PLEASE req S-TOKEN-PLEASE ind	Tokens, SS-user data
Minor Synchronization Point	S-SYNC-MINOR req S-SYNC-MINOR ind S-SYNC-MINOR resp S-SYNC-MINOR conf	Type, Sync-point serial number, SS-user data
Major Synchronization Point	S-SYNC-MAJOR req S-SYNC-MAJOR ind S-SYNC-MAJOR resp S-SYNC-MAJOR conf	Sync-point serial number, SS-user data
Resynchronize	S-RESYNCHRONIZE req S-RESYNCHRONIZE ind S-RESYNCHRONIZE resp S-RESYNCHRONIZE conf	Resynchronize type, Sync-point serial number, Tokens, SS-user data
U-Exception Reporting	S-U-EXCEPTION-REPORT req S-U-EXCEPTION-REPORT ind	Reason, SS-user data
P-Exception Report	S-P-EXCEPTION-REPORT ind	Reason
Activity Start	S-ACTIVITY-START req S-ACTIVITY-START ind	Activity identifier, SS-user data
Activity Resume	S-ACTIVITY-RESUME req S-ACTIVITY-RESUME ind	Activity identifier, Old activity identifier, Sync-point serial number, Old session connection identifier, SS-user data
Activity End	S-ACTIVITY-END req S-ACTIVITY-END ind S-ACTIVITY-END resp S-ACTIVITY-END conf	Sync-point serial number, SS-user data
Activity Interrupt	S-ACTIVITY-INTERRUPT req S-ACTIVITY-INTERRUPT ind S-ACTIVITY-INTERRUPT resp S-ACTIVITY-INTERRUPT conf	No parameters
Activity Discard	S-ACTIVITY-DISCARD req S-ACTIVITY-DISCARD ind S-ACTIVITY-DISCARD resp S-ACTIVITY-DISCARD conf	No parameters
Give Control	S-ACTIVITY-GIVE req S-ACTIVITY-GIVE ind	No parameters
Orderly Release	S-RELEASE req S-RELEASE ind S-RELEASE resp S-RELEASE conf	Result, SS-user data
U-Abort	S-U-ABORT req S-U-ABORT ind	SS-user data
P-Abort	S-P-ABORT ind	Reason

① BCS (Basic Combined Subset)

最も基本的なサービスを提供するサブセットで、カーネル、半二重または全二重FUからなり、SCごとに半二重ないしは全二重のいずれかが選択される。

② BSS (Basic Synchronized Subset)

カーネル、ネゴシエートリソース、半二重ま

たは全二重、小同期、大同期、再同期FUから成る。

③ BAS (Basic Activity Subset)

カーネル、半二重、タイプトデータ、ケーバビリティデータ交換、小同期、例外報告。アクレイビティ管理FUから成る。

現在、高位層プロトコルの標準化作業が継続中で、不確定な要素も多いがBCSでは同期関連のサービスを伴わないため、仮想端末やビデオテックスのような会話型通信ないしは簡便な全二重通信に使用されることを想定している。またBSSやBASはいずれも同期関連のサービスを含むが、BSSはファイル転送、ジョブ転送で使用されることを期待し、BASは、テレテックスおよびMHSで使用されることを想定している。

3.3 プロトコルの動作

セッションプロトコルの動作をセッションコネクション(SC)確立、データ転送、SCの解放の3フェーズに分けて説明する。

(1) SCの確立

本フェーズではSSユーザ間でSCを確立する。SSユーザからS-CONNECT reqを受けたSPM (Session Protocol Machine)は、まず、下位のトランスポートコネクション(TC)との関連付けを行い、TCを確立後、相手SPMにCN-SPDUを送信する。CN-SPDUを受信したSPMは、SSユーザに対してS-CONNECT indを通知する。

着呼SSユーザがSC確立を認める場合には、S-CONNECT resp (accept)をSPMに与え、その結果、AC-SPDUが送信される。またSC確立を拒否する場合には、S-CONNECT resp (reject)またはS-U-ABORT reqをSPMに与え、各々RF-SPDUまたはAB-SPDUが送信される。(図3参照)

SC確立時には双方のSPM間で、使用する機能単位、初期のトークン配置、初期の同期点順序番号、最大TSDU長(セグメンティングに関係する)が折衝される。

(2) データ転送

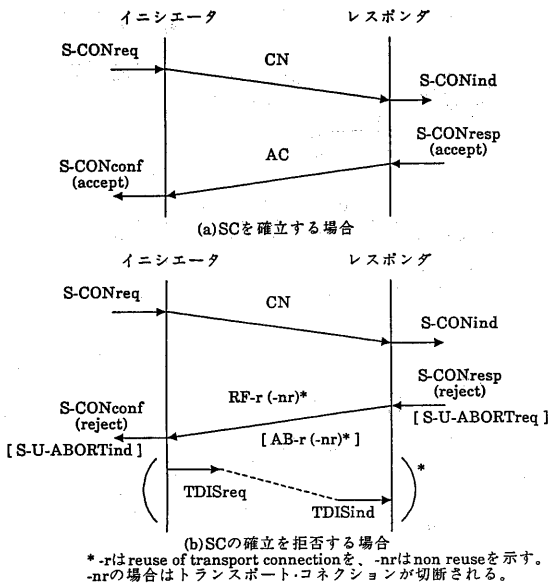


図3 コネクション確立シーケンス

本フェーズでは、確立したSC上でSSユーザからの要求に応じて、

- ① 普通、優先、タイプト、ケーパビリティ等のデータ
 - ② トークンの制御
 - ③ データの流れの中に小/大同期点を挿入する同期制御
 - ④ 何らかの誤りが生じデータが失われた場合の再同期制御
 - ⑤ アクティビティ制御
- 等のデータや制御情報を送受する。

ここで、データ転送の代表的なシーケンスについて説明する。普通データ転送では、SSユーザからのS-DATA reqにより、DT-SPDUで転送される。DT-SPDUにはセグメンティングが行われた場合にSSDUの先頭と終了を示す情報とユーザデータが含まれる。DT-SPDUを受信した場合は、S-DATA indとしてSSユーザに伝達される。DT-SPDUの転送には、半二重の場合はデータトークンの所有側がDT-SPDUを送信する権利をもつ。また、トークンの授受には、GT-SPDU、PT-SPDUを用いる。一方、全

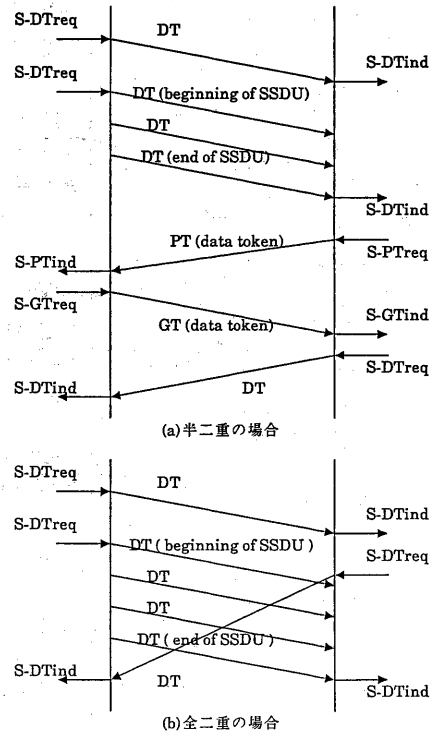


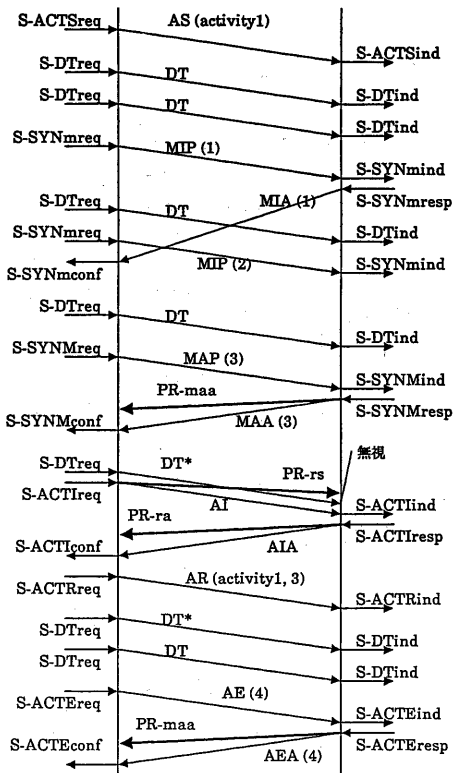
図4 普通データ転送シーケンス

二重の場合にはトークンには制御されず、普通データの両方向同時転送を行う。(図4参照)

また、アクティビティ管理、同期を含めたシーケンス例を図5に示す。

(3) SCの解放

本フェーズではSCの切断を行う。SSユーザ間でデータ転送が正常に完了した後S-RELEASE reqをSPMに与えFN-SPDUを送信し、相手SS-USERからのS-RELEASE resp (affirmative)によるDN-SPDUの受信による正常解放と、AB-SPDUの送受に伴い、データの紛失を伴う場合もある異常終了がある。また、ネゴシエトリリースFUが使用されている場合には、相手SPMからのFN-SPDUによる切断要求を拒否し、S-RELEASE resp (negative)によりNF-SPDUを送信し、SCを継続して使用できる。(図6参照)



注 S-ACTs: S-ACTIVITY-START, S-ACTI: S-ACTIVITY-INTERRUPT, S-ACTR: S-ACTIVITY-RESUME, S-ACTE: S-ACTIVITY-END, S-SYNm: S-SYNC-MINOR, S-SYMM: S-SYNC-MAJORを示す
 AS, ARで選ばれる activity1は Activity identifierである
 MIP, MAP, AR, AE等で選ばれる数字は同期点順序番号を示す
 ←トランスポート優先データ転送を示す
 *それぞれ同じデータを運ぶDT SPDUである

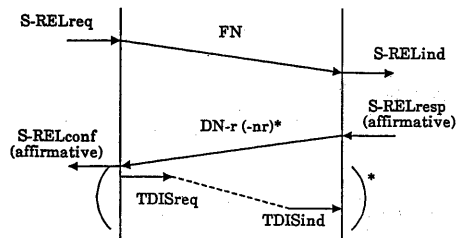
図5 アクティビティ管理、同期のシーケンス

4. OSIセッションレイヤ標準の実装

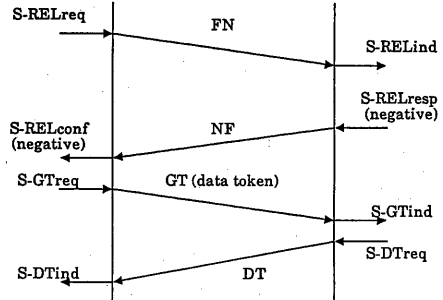
4.1 設計方針

OSIセッション標準の実装にあたっては、VAX11/780のVMS下で、既に開発済みのトランスポートプロトコルのクラス0、1、2、3のプログラムとインタフェースして動作することを前提としたが、その設計方針は以下の通りである。

- ① 可能なかぎり全仕様のセッションプロトコルを実現する。
- ② 異なる種類の上位レイヤプロトコルと同時にインタフェースし、また複数本のSCを同時に張ることができるようにする。
- ③ 将来の拡張、修正が容易なように、またシステムに合わせて必要な機能単位のみをくり出して実



(a) SCを解放する場合
 *-rはreuse of transport connectionを、-nrはnon reuseを示す。
 -nrの場合はトランスポート・コネクションが切断される。



(b) SCを解放しない場合

図6 SC解放シーケンス

装できるように、FUに準拠したモジュール化を行い、状態遷移表については一元的に管理する。

- ④ 上位レイヤとのインタフェースは、セッション層のサービスプリミティブをPDUに近い形式でフォーマット化して用いる。
- ⑤ C言語で記述して移植性を高める。

4.2 ソフトウェア構成

セッションプログラムに着目した網内HOSTのソフトウェア構成を図7に示す。セッションプログラムは

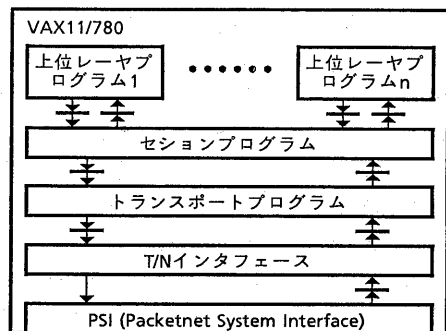


図7 網内HOSTのソフトウェア構成

VMSの下で1つのプロセスとして実装され、トランスポートプログラムを実行するプロセスと、複数の

上位レーヤプロセスとメールボックスを介してインタフェースする。現在は最大8個の上位レーヤプロセスとインタフェースできるようにしている。

(1) セッションプログラムのモジュール構成

作成したセッションプログラムはFUに準拠したモジュール化を行い、図8に示すように、制御モジュール、機能単位モジュール群、共通モジュールという3階層のモジュール構造をとっている。各モジュールの機能概要は以下の通りである。

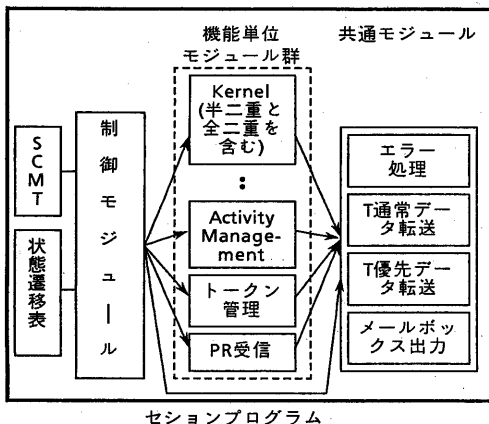


図8 セッションプログラムのモジュール構成

① 制御モジュールは、メインプログラムを含むモジュールであり、

(i) 上位およびトランスポートのプロセスからのプリミティブの読みこみ、解析

(ii) 状態遷移表の管理

(iii) 機能単位モジュールまたは共通モジュールのコール

(iv) SC毎に割り当てたSCMT (Session Connection Management Table)を用いた、複数のSCの管理を行う。

SCMTはSC毎にその確立/解放にあわせて確保/解放される。SCMTには

- ・SSAPアドレス、TSAPアドレス、メールボックスのチャンネル番号等の上位・下位レーヤとの対応をとるための情報

- ・そのSCで使用するFU、トークンの配置、SCの状態、設定した同期点の順序番号等、そのSCを特徴付けるための情報

が蓄えられる。

② 機能単位モジュールは、各FUに関連するサービスプリミティブとSPDUに対する処理を実行するもので、原則的にFU毎に作られている。これらの内、カーネルに半二重と全二重の機能を含ませている。また、トークンは複数のFUで共通に使用されるため、新たにトークン管理モジュールを設け、S-TOKEN-GIVE、S-TOKEN-PLEASEプリミティブとGT、PT-SPDUの処理を行う。PR-SPDUはトランスポート層の優先データ転送サービスが使用可能な場合に、小同期、大同期、再同期、アクティビティ管理のFUで共通に用いられ、PR-SPDUを受信した時点ではそれがどのFUで用いられたものかが分からない。従ってPR-SPDUの受信処理を一元的に行うための機能単位モジュールを設けた。

③ 共通モジュールは、制御モジュール、各機能単位モジュールが共通に使用する処理の集まりで、プロトコルエラーの処理、SSユーザからのプリミティブのエラー(ユーザエラー)の処理、SPDUの転送を行うために必要なトランスポートのサービスプリミティブを作成しメールボックスに出力する処理等を行う。

(2) 状態遷移表の構成と処理の流れ

セッションプロトコルの状態遷移表は状態数が29、入力数が77で、更に同一の状態と入力の組に対しても条件により異なる動作があり、複雑なものとなっている。このため、遷移表の保守・変更、必要最小限のFUのくくり出しが容易な様に、状態遷移表を次の様な構成にする。

① 遷移表は制御モジュールが一元的に保持し、遷移表の参照も制御モジュールが行う。

② 遷移表の保守・変更を容易にするためには、遷移表に記述される内容を少なくする必要がある。そこで、遷移表の各エントリには、受信したプリミティブやSPDUを処理すべき機能単位または共通モジュールの名と、各モジュールに処理の内容を指示する処理コードだけが記入されている。

制御モジュールは、受信したセッションまたはトランスポートのサービスプリミティブを解析し、状態遷移表を参照する。その後、遷移表の該当するエントリに示されたモジュールに受信したプリミティブ/SPDUと処理コードを渡す。一方、機能単位または

共通モジュールは受けとった処理コードが示す条件判断、状態の遷移、プリミティブやSPDUの出力、順序番号等に対応する変数の更新を行う。

図9にMAP-SPDUを受信した場合を例にとり、その処理の流れを示す。状態STA713(データ転送中の状態)でMAP-SPDUを受信した場合、制御モジュールは状態遷移表を参照し、対応するエントリに規定されているモジュール名(MAJOR SYNC)と処理コード(MJ 014)を読み、MAJOR SYNC機能単位モジュールをコールする。MAJOR SYNC機能単位モジュールでは、はじめに処理コードによる場合分けを行う。MJ 014 に対しては

- ・トークンの配置、MAP-SPDUの同期点順序番号に関するチェックを行う。
- ・正しい場合はS-SYNC-MAJOR indを発行し、同期に関連する変数を更新し、状態STA10A(SSユーザからS-SYNC-MAJOR respを待つ状態)に遷移する。
- ・正しくない場合は共通モジュールにあるプロトコルエラーを処理する関数P-ERRORをコールする。

という処理を行う。

一方、STA10AでMAP-SPDUを受信した場合、MAP-SPDUを続けて受信したことになり、プロトコルエラーの処理が必要である。従って状態遷移表

の対応するエントリにはモジュール名として共通モジュールの内の関数P-ERRORが記されており、制御モジュールは直接この関数をコールする。P-ERRORでは、AB-SPDUまたはER-SPDUを出力しそれぞれに対応する状態の遷移を行う。

(3) サービスサブセットのシミュレーションプログラム

セッションサービス定義に規定されているBCS、BSS、BASの各サービスサブセットに対して、これらをシミュレートする上位プログラムを作成した。

BCSとBSSのシミュレーションプログラムでは、ファイルとして蓄えられたデータの転送と、コンソールから入力されたテキストの転送が可能である。更にBSSでは同期再同期をシミュレートするため、ファイル転送の場合には

- ・異なるファイルの転送は大同期点により区切る。
- ・小同期点は、ファイルの中に設けられた制御コードによって設定する。
- ・ファイルへのデータの書き込みは各同期点毎に行う。
- ・ファイル受信中にコンソールから受信の一時停止、再開を指示できる。受信再開の指示により再同期(リスタート)を行う。

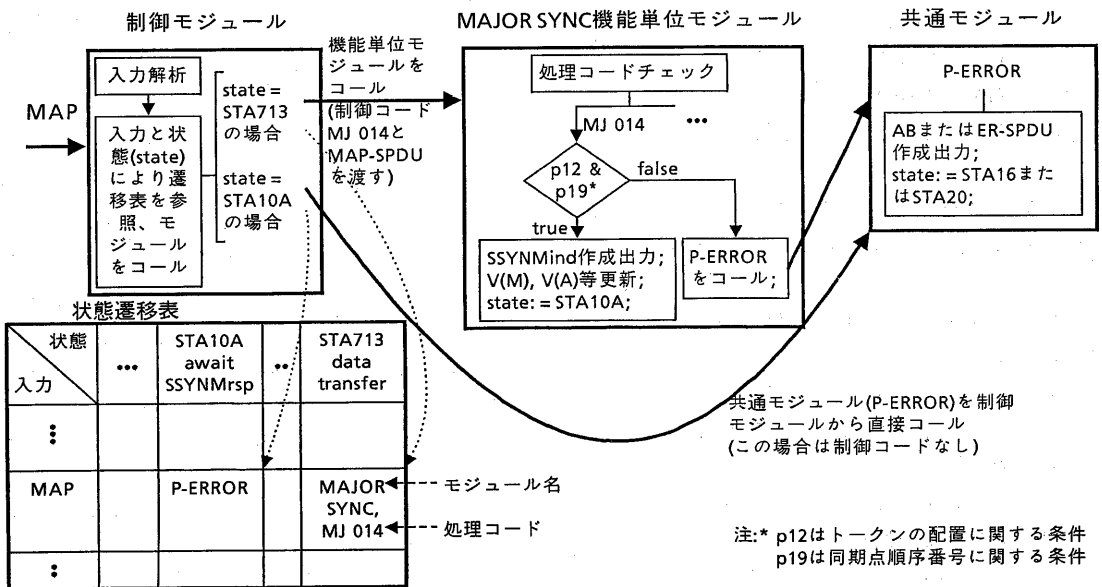


図9 MAP-SPDUを受信した場合の処理の流れ

等の機能を具備している。一方、テキスト転送の場合には

・コンソールからのコマンドにより小同期点を設定する。

・テキスト転送でも受信したデータをファイルに書き込むこととし、この書き込みは小同期点毎に行う。

・送信側からの再同期をシミュレートするために、最新の同期点のあとで転送したテキストをキャンセルするためのコマンドを用意し、このコマンドにより送信側から再同期(リスタート)を行う。

等の機能を具備している。

またBASのシミュレーションプログラムではファイル転送のみを行い、1つのファイルの転送をアクティビティに対応させ、小同期点はファイルの中に設けられた制御コードによって設定する。更にCCITT勧告T.62(1980年版CCITT勧告ではS.62)をシミュレートする場合は、T.62に特有のService IdentifierやNon-Basic Session Capabilities等のパラメータの取り扱いなどを指定した動作指示ファイルを用意した。但し、セッションプログラムでT.62特有のパラメータを扱うためには、後述するようにセッションプログラム自身も改修する必要がある。

5. 結果と考察

OSIセッション標準を実装し、X.25 パケット網を介して折り返し試験を行い以下のような結果と考察を得た。

- (1) セッションプログラムは簡易的なデータ転送プログラムや、BCS、BSS、BASシミュレーションプログラムを用いて、正常に動作している。予備的な実験ではあるが、100K byteのファイルをX.25 パケット網折り返し(1回線で行き帰り別々の論理チャンネルを割り当てる)で転送した場合、回線速度4800bps、K=4、W=3の条件で2Kbpsのスループットを確認している。
- (2) 表3に、C言語で記述したソースプログラムの実効行数(コメント文と空白行を除いた行数)を各モジュールごとに示す。セッションプログラムの総実効行数は約20K行である。これらの数値は、プログラムの記述方法で若干変動するが、セッションプログラムの規模を把握できる。

表3 SPソースプログラムの実効行数

モジュール	規模(実効行数)
制御モジュール (状態遷移表を含む)	4614
Kernel Half duplex Duplex	4760
Negotiated release	219
Expedited	389
Typed data	372
Capability data exchange	421
Minor synchronize	1015
Major synchronize	1028
Resynchronize	2314
Exceptions	741
Activity Management	2581
トークン管理	909
PR 受信処理	152
共通モジュール	299
合計	19814

- (3) BAS サブセットをシミュレートする場合、テレテックス用のT.62と通信を想定すると、T.62固有のパラメータの取り扱いに関して、セッションプログラムの内部処理を以下のように変更する必要がある。

① 上位レーヤがT.62特有のパラメータを送信する場合はサービスプリミティブのユーザデータとしてこれらのパラメータをセッションプログラムに渡す。セッションプログラムではこれらのパラメータを、T.62の定めるフォーマットに従ってSPDUに展開する。

② 逆に、T.62特有のパラメータを持つSPDUを受信した場合は、対応するサービスプリミティブのユーザデータに入れる。

- (4) 今回作成したセッションプログラムは、OSIセッションプロトコルの内、拡張コンカティネーション以外のすべて機能を実装している。拡張コンカティネーションが実装できなかった理由は、コンカティネーションを行うタイミングをとるのを省いたためである。

- (5) 今回作成したプログラム構成は、保守、変更が容易で、例えば必要なFUのみををくり出したプログラムを作成するには、

① 状態遷移表は一切変更を加えず、

② 使用しない機能単位モジュールを、セッションプリミティブに対しては共通モジュールにあるユーザエラーを処理する関数をコールし、また

SPDUに対しては関数P-ERRORをコールするよ
うに変更する
ことで対処できる。

6. あとがき

本プログラムの実装に伴う検討過程で多数の
CCITT寄書、ISO日本寄書が作成されており、OSI
セッション・レーヤの勧告・標準作成に役立ってい
る。今後は、これまで実装したトランスポート、セ
ッションプログラムの処理効率の向上を図るととも
に、OSI各種プロトコルの検討・実装を進めたい。最
後に日頃御指導頂く、KDD研究所鍛冶所長、野坂副
所長、深田次長、小野情報処理研究室長に感謝する。

参考文献

- [1] CCITT Rec. X.215, X.225 March 1984 (ISO DIS 8326, 8327).
- [2] 鈴木他 “OSI参照モデルにおけるトランスポート、セッション・レーヤについて,” 情報処理学会分散処理システム研究会, 19-4, July 1983.
- [3] 鈴木, 加藤 “OSIトランスポートプロトコルのインプリメントと製品検証,” 情報処理学会分散処理システム研究会, 22-9, May 1984.
- [4] 鈴木, 加藤 “OSIセッション・プロトコルのインプリメント,” 情報処理学会第29回全国大会, 2H-3, Sept. 1984.