解　説

形式記述技法の OSI プロトコルへの適用†

大　原　康　博††

1. はじめに

OSI のプロトコルおよびサービスの規定は，従来自然言語によって行われ，部分的に状態遷移図・表で補われてきた．しかし最近形式記述技法（以降 FDT: Formal Description Technique）の開発・整備とともにこれらの規定が徐々に FDT で記述されつつあり，従来の自然言語による規定の曖昧さを補っていくことが期待されている[1]．

自然言語で規定された OSI の仕様を FDT で記述するためには，その仕様に含まれる概念が FDT で記述されなければならない．OSI の規定は一般に，アーキテクチャ，サービスおよびプロトコルの3要素からなっており，FDT にはこれらの内容が記述できる能力が要求される．

（1） アーキテクチャの記述に対しては，サービス・プロトコルを記述する前提となる環境や基本概念などを記述できるような抽象度の高い記述性が要求される．

（2） サービスの記述に対しては，OSI の各層が上位層に提供する機能集合を記述するため，プロトコル処理要素（エンティティ）の内部動作を隠蔽した記述性が要求される．

（3） プロトコルの記述に対しては，エンティティの機能・動作のすべてを記述できる能力が要求される．特に，プロトコルデータの形式やタイミング条件などの記述が重要になる．

ここでは，上記の要求を満たす FDT によりこれまで記述された具体的な OSI プロトコル，サービスの標準化状況およびいくつかの記述例について述べる．まず，OSI 参照モデル（アーキテクチャ）の主要概念の FDT 記述法を述べた後，代表的な FDT による記述例として，LOTOS を用いた OSI セションプロトコルの記述を示す[2]．

† Application to OSI Protocol Specifications of Formal Description Techniques by Yasuhiro OHARA (NTT Telecommunications Service Support Headquarters, Operations Systems Development Center).

†† NTT 電話事業サポート本部・オペレーションシステム開発センタ

2. FDT 規定における規定内容と規定状況

現在，OSI 仕様のうち FDT により記述されているまたはされつつあるものは，以下に述べるいくつかのサービス定義とプロトコル仕様のみである．アーキテクチャ記述については，OSI 参照モデル（ISO 7498）の FDT 記述は行われていないが，各サービス／プロトコルの記述にあたって必要な概念などの記述がそれらの仕様記述の中で行われている．各種 OSI サービス/プロトコルの FDT 記述状況は，以下のとおりである．今後，ODP（開放型分散処理）用サービス／プロトコルなどが記述される方向に検討が進むと考えられる．

・応用層（レイヤ7）のサービス／プロトコル仕様として，トランザクション処理の主要部分の動作記述が，Estelle（状態遷移モデル）および LOTOS（時系列動作モデル）により行われている[3),4)]．

・セション層（レイヤ5）およびトランスポート層（レイヤ4）のサービス/プロトコル仕様として，それらの動作手順が，起こり得るイベントの順序を宣言的に記述する時系列動作モデルに基づいて，LOTOS により記述されている．典型的な例として，セション層プロトコルの記述を，4. で述べる[5),6)]．

・データリンク層（レイヤ2）については，ローカルエリアネットワーク（LAN）プロトコルの，媒体アクセス法（MAC）の動作手順が，プログラミング言語 Pascal を用いて形式的に記述されている．具体的には，状態遷移モデルに基づきフレームの送受信処理における遷移動作が，Pascal の手続き（Procedure）および関数（Function）により記述されている[7)]．

なお，応用層，プレゼンテーション層におけるプロトコルデータ（PDU など）の記述法としては，ASN.1 が使用されている．

3. 主要な OSI アーキテクチャの FDT 記述

OSI 参照モデルで規定された概念のうち，サービス／プロトコル仕様の記述で必要な主なものについ

て，記述法を示す．なお，FDT として SDL，LOTOS，および Estelle を用いた記述例を示す．

OSI 参照モデルでは，任意の応用プロセス間の通信を可能とするため，物理媒体の制御から仮想資源へのアクセス制御までの通信に関する各種のプロトコルを，機能の共通性・独立性の観点から7つに分割し，動作する順序に階層化している(7層モデル)．また，各層(〈N〉層)に共通の構成要素とその概念を定義している．主要なものとして，〈N〉エンティティ(プロトコルの実行主体)，〈N〉プロトコル(〈N〉エンティティが相互に通信するための手順や制御情報などを定めたもの)，〈N〉サービス(〈N〉層が〈N+1〉層に提供するコネクション設定，データ転送などの機能)，〈N〉SAP(〈N+1〉エンティティが〈N〉サービスを受けるためにアクセスする論理的な口)，〈N〉コネクション(〈N+1〉エンティティ間通信のために作られる論理的な通信路)，などが定義されている(図-1)[8),9)]．これらのうち，以下では〈N〉SAP およびサービスプリミティブ〈N〉サービスの各要素)について，FDT による記述法を例示する．

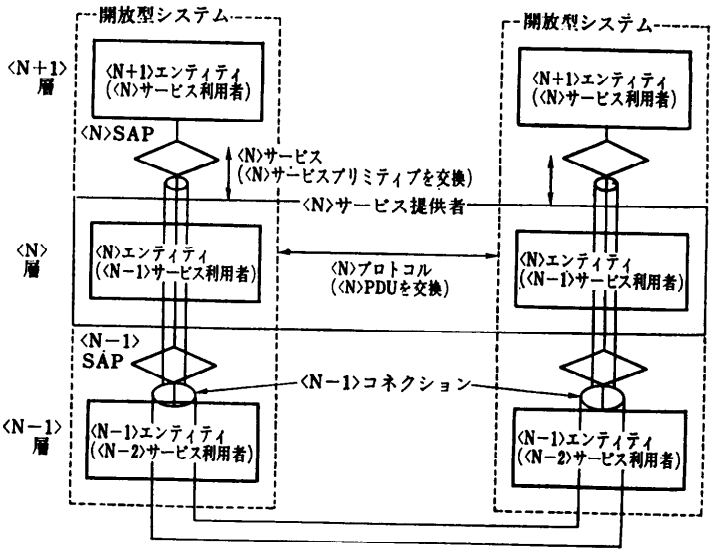

図-1 OSI 参照モデルにおけるプロトコル層の構造

3.1 サービスアクセス点 (SAP)

サービスアクセス点は，〈N〉層エンティティと〈N+1〉層エンティティとの間で OSI サービスを授受する箇所であり，サービスプリミティブがやりとりされる．このため，どの FDT でも相互動作点 (Interaction point) という概念として表現される．

(1) SDL による記述

ブロック間でシグナル (SIGNAL) を交換するチャネル (CHANNEL) として表される(図-2)．

Sap はチャネル名で出力 (OUTPUT) の中でシグナルを指定するために用いる．Event 1 などはこの Sap で交換されるシグナル名である．FROM, TO で指定された ServiceUser, ProtocolEntity は Sap を介して通信する相手同士を示し，それぞれサービスユーザとプロトコルエンティティである．

(2) LOTOS による記述

次の形をしたイベントとして表現される(図-3)．

Sはサービスにアクセスするための通信が行われる場所（ゲート）を示し，Sap はその識別子であり，通常 SAP アドレスが入る．この Sap はある型 (SapType) のインスタンスを表すもので，型 SapType は，抽象データ型により定義される(図-4)．

```
CHANNEL  Sap
  FROM  ServiceUser  TO  ProtocolEntity
  WITH  Event 1, Event 2, ... ;
  FROM  ProtocolEntity  TO  ServiceUser
  WITH  Event 3, Event 2, ... ;
ENDCHANNEL;
```

図-2 SAP の SDL による記述例

```
S ! Sap !...
```

図-3 SAP の LOTOS による記述例

```
type  SapType  is
  sorts  SapSort
  opns  BaseSap: -)SapSort
        NextSap: SapSort  -)SapSort
endtype
```

図-4 SAP データの LOTOS による抽象データ型定義例

(3) Estelle による記述

外部プロセスとの相互動作点として表される．もっとも単純な場合は一つの相互動作点となるが，一般には SAP の中に複数の端点 (Endpoints) を含むので相互動作点の配列 (Array) として表される．相互動作点はモジュール定義の中で記述される(図-5)．

Mはプロセス型のモジュールで ServiceProvider としての役割をする SAP をもつ．SAP 自身はチャネ

```
module M process;
  ip SAP: SAP (Provider);
end;
channel SAP (User, Provider);
  by User: ConReq, ConResp;
  by Provider: ConInd, ConConf;
```

図-5 SAP の Estelle による記述例

```
OUTPUT Sp(Sap, Ep, DataInd(...));
```

図-6 サービスプリミティブの SDL による記述例

```
NEWTYPE UserData OctetString
ENDTYPE;
```

図-7 サービスプリミティブパラメータの SDL による記述例

```
S ! Sap ! Ep ! CONNECTreq(Addr 1, Addr 2, QOS)
```

図-8 サービスプリミティブの LOTOS による記述例

```
type BasicSP is Address, OctetString
  sorts SP
  opns CONNECTreq, CONNECTind: Address, Address, QOS ->SP
       DISCONreq, DISCONind: ->SP
       DatReq, DatInd: OctetString ->SP
endtype
```

図-9 サービスプリミティブパラメータの LOTOS による記述例 (その1)

```
type Address is Boolean
  sorts Address
  opns SomeAddress: ->Address
       AnotherAddress: Address ->Address
       _eq_, _ne_: Address ->Bool
endtype
```

図-10 サービスプリミティブパラメータの LOTOS による記述例 (その2)

```
channel SAP(user, provider);
  by User:
    CONNECTrequest(Addr 1, Addr 2, QOS);
    CONNECTresponse(Addr 1, Addr 2 QOS);
  by provider:
    CONNECTindication(Addr 1, Addr 2, QOS);
    CONNECTconfirm(Addr 1, Addr 2, QOS);
```

図-11 サービスプリミティブおよびパラメータの Estelle による記述例

ル定義の中でその役割や交換されるイベントを定義する.

3.2 サービスプリミティブおよびそのパラメータ

SAP を通じて〈N〉エンティティと〈N+1〉エンティティ間で交換されるサービスプリミティブは, どの FDT でもイベント (あるいは信号) として表現される. またサービスプリミティブに含まれるパラメータの記述は, データの定義として記述される.

(1) SDL による記述

シグナルのインスタンスとして表される (**図-6**).

サービスプリミティブ Sp は, SIGNAL により定義されるシグナル型のインスタンスで, サービスアクセス点 Sap, 端点 Ep およびプリミティブ自身 DataInd が指定されている. Sap, Ep などを陽に表示する必要がない場合は, プリミティブ自身のみを示し OUTPUT DataInd (UserData); のように記述することができる. サービスプリミティブのパラメータは, 別に NEWTYPE により型宣言されるデータとして記述される. たとえば, UserData パラメータの場合, UserData がオクテット列型データとして定義される (**図-7**).

(2) LOTOS による記述

イベント列としてサービスプリミティブの送信・受信などを記述する (**図-8**).

CONNECTreq はサービスプリミティブの送信を示す. これは抽象データ型により表現されるサービスプリミティブ定義の中で与えられるものである.

たとえば, **図-9** のように, サービスプリミティブ集合 BasicSP が抽象データ型として定義され, その中で CONNECTreq が定義される. サービスプリミティブのパラメータは, 同様に抽象データ型として定義される. この例でいえば, CONNECTreq のパラメータ Addr 1, Addr 2, は次のように定義される (**図-10**).

(3) Estelle による記述

チャネル定義で定義される相互作用として記述される. サービスプリミティブパラメータは, それら相互作用のパラメータとして記述される (**図-11**).

CONNECTrequest, CONNECTindication などがサービスプリミティブで, サービスプリミティブのパラメータは, Addr 1, Addr 2, QOS などである.

4. OSI サービス/プロトコルの規定例

具体的な OSI サービス/プロトコルの記述例として, 現在もっとも整備された形で FDT 記述がなされている, セション層プロトコルの LOTOS による記述を説明する. これは, もとの自然言語による仕様とほぼ同等な内容を含んでおり, 現在国際標準に準じる技術レポート (TR) として, 標準化されている.

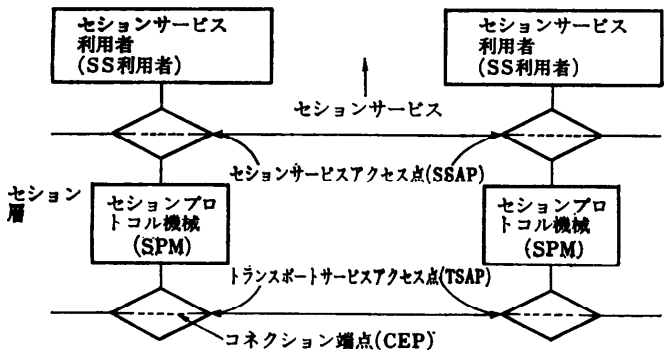

図-12 セションプロトコルのモデル

4.1 OSI のセッション層プロトコル規定

OSI のセション層プロトコルは，基本コネクション型セションプロトコル仕様（バージョン1）が ISO 8327 として 1985 年に ISO で規格化された．なお国内ではこれを受けて ISO 8327 に含まれていた誤りを修正し JIS X 5202 (1987) が作成されている．

セションプロトコルは OSI 参照モデル上，下位のトランスポートサービスを利用し上位層にセションサービスを提供するものとして位置づけられる．セションプロトコルの動作は，セションプロトコル機械 (SPM) としてモデル化され，セションサービスアクセス点 (SSAP) をとおしてセション層の利用者 (SS 利用者；実際にはプレゼンテーション層以上のエンティティを指す）と通信し，トランスポートサービスアクセス点 (TSAP) をとおしてトランスポート層と通信する (図-12)[8)~11)]．

このモデルにおいて，一方の SPM は SS 利用者からの要求をセションサービスプリミティブとして受け取り，これを相手側 SPM にセションプロトコルデータ単位 (SPDU) として伝える．これらプロトコルの交換は，トランスポート層が提供するトランスポートサービスプリミティブを用いて行われる（トランスポート層ではこれをトランスポートプロトコルデータ単位 (TPDU) として相手に伝える）．コネクション型の仕様では，SPDU の交換のため SPM 間のセションコネクションが設定される．トランスポート層にもトランスポートコネクションが設定される．セションコネクションは一対の SSAP 間で複数可能で，SSAP 内に定義されるコネクション端点でこれらが識別される．

セション層の機能は，SS 利用者間での会話管理，データ流れの同期制御，誤り通知，および誤り回復時の再同期制御，などからなっており，これらに対応したプロトコルが規定されている．主要なものを以下に示す．

（1） コネクション確立フェーズ

SS 利用者からのコネクション確立要求に応じ，相手 SS 利用者との間にコネクションを確立するため，次の機能が実行される．

（a） セションアドレスをトランスポートアドレスに写像し，トランスポートコネクションの確立を行う．この場合，すでにトランスポートコネクションが張られており利用できれば，新たに確立せずに再利用する．

（b） SS 利用者から指定されたサービス品質パラメータから，必要なトランスポートサービス品質を選択する．

（c） セションコネクションで用いる機能（全二重／半二重，同期・再同期など）やパラメータ（セション上の最大データ長，初期のデータ送信権など）をネゴシエートする．

（d） 個々のセションコネクションを識別する．(コネクション識別子)

（2） データ転送フェーズ

セションコネクションが確立されると，SS 利用者間でデータ（セションサービスデータ単位：SSDU）の転送ができるようになる．セション層はこれを SPDU として転送する．またこの転送を支援するため，コネクション確立フェーズで選ばれた機能が実行される．主な機能として

（a） 普通データ転送：送信権・フロー制御の制約を受ける普通データ SSDU への分割および相手側 SPM での組立，二つ以上の SPDU を同時に転送するための連結／分離を行う．

（b） 優先データ転送：送信権・フロー制御の制約を受けない優先データ SSDU の転送を行う．

（c） トークンの譲渡・要求：データの送信権，サービスの起動権などの属性を表すトークンを，一方の SS 利用者から他方の SS 利用者へ渡す／要求するための制御を行う．

（d） 同期点制御：データの再送や途中での送達確認を行ったりするために，チェックポイント（同期

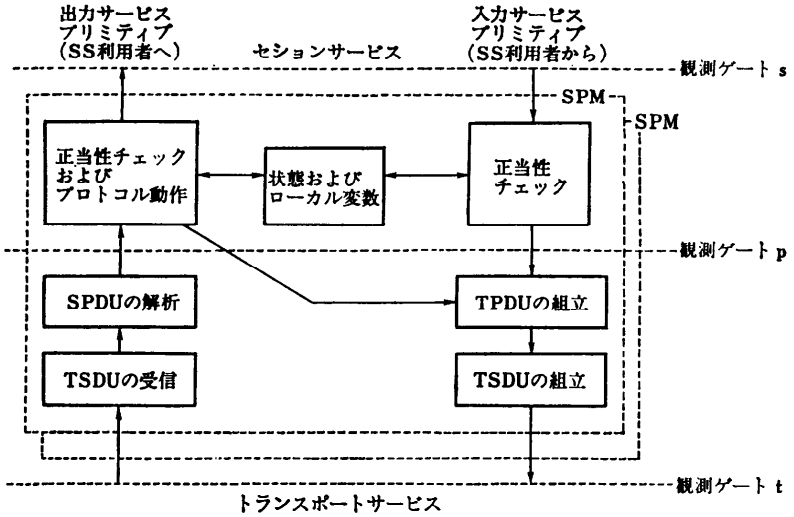

図-13 形式記述のためのセション層プロトコル処理機械の動作モデル

点）の設定などを行う．

（e） 例外報告：SS 利用者やプロトコルエンティティにおいて検出した誤りを，相手に通知する．などがある．

（3） コネクション解放フェーズ

セションコネクションを正常または異常に解放する機能である．

（a） 正常な解放は，すべてのデータ転送が終了した後 SS 利用者からの要求で解放され，データの損失はともなわない．

（b） 異常解放は，セション層以下の機能の異常または利用者からの中断要求により，データ転送の途中でコネクションを解放するもので，データの損失をともなう．

4.2 形式記述のためのセション層動作のモデル化

上に述べたセションプロトコルを LOTOS を用いて形式記述する場合，いわゆる制約指向型の記述 (constraint-oriented specification) が基本的に採用されている．具体的には，

1. セションプロトコル機械の動作（プロセス：SetOfSPM)☆，

2. セションサービスとの境界点でセションコネクションの正しい割当てに関するローカルな条件（プロセス：SCIdAllocation, SCSupport),

3. トランスポートサービスとの境界点でトランスポートコネクションの正しい割当てに関するローカルな条件(プロセス :TCEIdentification, TCAcceptance, TBackpressure) に分解され，これらおのおのの動作条件の集まりとして記述されるスタイルを取っている．条件の集まりは LOTOS の並列合成として記述される．1. は図-13 のモデルで示されるセションプロトコル機械 (SPM) の集合として，2. はセションサービス，3. はトランスポートサービスの中で記述される．

4.3 LOTOS によるセション記述内容の概要

セションプロトコル仕様の形式記述（以後，セション FD) は，基本的には ISO セションプロトコル仕様 (ISO 8327) の状態遷移動作を形式的に記述したものであるが，自然言語記述に比べ以下のようないくつかの特徴をもっている．

（a） ISO 8327 の状態遷移表では，SPDU/SSDU が扱われるデータ単位であるが，セション FD ではより詳しくそれを構成するオクテット単位まで記述するため，各フィールド単位の処理にともなう状態遷移を厳密に規定することができる．（たとえば，あるSPDU のあるパラメータフィールドでエラーが検出された場合，どういう動作をするか，など）

（b） ISO 8327 の状態遷移表では，SPDU 連結（コンカティネーション）や SSDU 分割（セグメンティング）などが明確に記述されていないが，セション FD ではこれらを記述する．このため，複数の関連する SPDU 間で起こる異常時などの処理が，明確に規定できる．（たとえば，連結された SPDU の中にエラーを検出した場合，残りの正しい SPDU の処理をどうするか，など．）

☆ 括弧内に示すプロセス名は，ISO の TR で用いられているもので，4.4 に記述例を示している．

```
specification SessionProtocol [s, t] (spei: SPEImplementation): noexit
  library Boolean, Elements, Set, String, NatRepresentations, OctetString,
          NaturalNumber, Bit, FBoolean, Octet, DecNatRepr, BitString, BitNatRepr,
          DecString, DecDigit (*1*)
  endlib
behaviour
  ((SCIdAllocation [s]||SCSupport [s]) (*2*)
    |||
    (TCEIdentification [t]||TCAcceptance [t]||TBackpressure [t]) (*3*)
  )
  ||SetOfSPM [s, t] (spei) (*4*)
where
  process SetOfSPM [s, t] (spei: SPEImplementation): noexit := (*5*)
    SPM [s, t] (spei)|||SetOfSPM [s, t] (spei)
  endproc
```

図-14 セション FD の全体概要

(c) ISO 8327 の状態遷移表では，不正データの受信時の処理が記述されておらず，本文中に分かれて記述されている．セション FD では状態遷移記述の中で明確に記述しており，網羅的に規定できる．

(d) ISO 8327 の状態遷移表では，SPDU とトランスポートサービスプリミティブ／セションサービスプリミティブとの対応関係など，レイヤ間にまたがるデータの相互関係が，記述されていないが，セション FD ではこれを記述している．(たとえば，セション利用者からのセションサービスプリミティブが，どの SPDU にマッピングされ送られるか，など)

これらの特徴をもつセション FD は，**4.2** に述べたモデルを用いて，次のように形式記述される．以下の解説で (＊n＊) は，LOTOS ではコメントとして解釈され後の説明で参照するものである (**図-14**)．

セション FD の中で扱うデータは，あらかじめ用意された抽象データ型の集まり，Boolean, BitString などにより表現でき，これらが library として参照される (＊1＊)．

セションの動作は，behaviour 節で示されるように，

- セションサービスの境界で成立する条件 (＊2＊)，すなわちセションコネクションの割当てを行うプロセス (SCIdAllocation) とセションコネクションの資源をチェックするプロセス (SCSupport) で示される制約条件が両立する動作，
- トランスポートサービスの境界で成立する条件 (＊3＊)，すなわちトランスポートコネクションの識別を行うプロセス (TCEIdentification) とトランスポートコネクションの受入れをチェックするプロセス (TCAcceptance) とトランスポート層でのフロー制御を行うプロセス (TBackpressure) の両立する動作，
- セション層機械の動作 (＊4＊)，が同期する (同時に成立する) 条件として記述される．特に中心となるプロセス SetOfSPM は，各セションコネクションに対する動作を扱う SPM の集まりとして表される (＊5＊)．各 SPM は，さらにセションサービスプリミティブ (SSP) の順序関係を制御するプロセス (SCEPs) (＊6＊)，セションの状態遷移動作を制御するプロセス STPM (＊7＊)，トランスポートサービスプリミティブ (TSP) の順序関係を制御するプロセス (TCEPs) (＊8＊)，が互いにセションサービス境界点 (s ゲート) とトランスポートサービス境界点 (t ゲート) で同期する条件として記述される (**図-15**)．

プロセス SCEPs は，ある一つのセションコネクションについての動作 (発呼側と着呼側での SSP の可能なシーケンス) を示し，コネクションが開放されると別のセションコネクションを確立して動作するか，すべて終了するかのどちらかの動作をする (＊9＊)．同様にプロセス TCEPs は，トランスポートコネクションでの TSP の動作を示す (＊10＊)．

プロセス STPM は，ISO 8327 の状態遷移表に基本的に対応する動作を表すが，これは SPDU と SSP, TSP の間に成り立つ関係として記述される．すなわち，SSPPDU (＊12＊) は SSP とそれから引き起こされる SPDU との関係 (例：S-CONNECTrequest と CN SPDU) を，SPDUTSP (＊13＊) は SPDU とそれから引き起こされる TSP との関係 (例：CN SPDU と T-CONNECTrequest) を，SPDUConstraints (＊14＊) は SPDU 間の関係を，STSP (＊11＊) は SPDU が介在しない SSP と TSP の関係 (例：T-DISCONNECTindication が S-P-ABORTindication を引き起こすこと) を示す．これらは，s ゲート，t ゲートおよび SPDU が通過する内部ゲート (p ゲート) での同期関係として記述されている．

```
Process SPM[s, t](spei : SPEImplementation) : NOEXIT :=
  (SCEPs[s]  (*6*)
    |[s]|
   STPM[s, t](spei)  (*7*)
    |[t]|
   TCEPs[t]  (*8*)
  )≫SPM[s, t](spei)
  where
   process SCEPs[s] : exit :=
    (SCEP[s](Calling) [ ] SCEP[s](Called))  (*9*)
     ≫(exit [ ] SCEPs[s])
    endproc
   process TCEPs[t] : exit :=
    TCEP[t](Calling) [ ] TCEP[t](Called)  (*10*)
   endproc
  process STPM[s, t](spei : SPEImplementation) : exit :=
   (STSP[s, t]  (*11*)
    ||
   hide p in (SSPPDU[s, p]  (*12*)
              |[p]|
              SPDUTSP[p, t]  (*13)
              |[p]|
              SPDUConstraints[p](spei)  (*14*)
             )
  ) [> exit
  endproc
endproc
```

図-15 セションプロトコル機械 (SPM) の記述例

```
type SessionAddress is Boolean
sorts SAddress
opns
  someAddress :->SAddress
  AnotherAddress : SAddress->SAddress
  _eq_, _ne_ : SAddress->Bool
eqns forall a, al : SAddress
    ofsort Bool
  a eq a =true ;
  someAddress eq AnotherAddress(a)=false ;
  AnotherAddress(a) eq someAddress=false ;
  AnotherAddress(al) eq AnotherAddress(a)=al eq a ;
  al ne a =not(al eq a) ;
endtype
```

図-16 セション FD で用いられるデータの記述例

これらに示したように，セションプロトコルの LOTOS 記述は PDU 単位の動作をプロセスとして選びそれらの間に成り立つ条件を積み上げていく方法で行われる．したがって，段階的にプロセス動作が詳細化され，分かりやすくかつ拡張性の高い表現法になっていることが分かる．これらのプロセスの中で参照されるデータは，次のように抽象データ型定義として，汎用的に表される．たとえば，たびたび使われるセションアドレス (SAddress) の定義は，以下のように定義される (**図-16**)．

5. あとがき

本解説では，FDT の具体的応用法として，OSI のサービス／プロトコル仕様をとりあげ，主要部分についての記述例を示した．これらは，セション層などを除き現在まだ不完全，未完成で今後より詳細記述が行われていく予定である．また，形式記述されたものは，自然言語で書かれた仕様に対する付属書（付録）として扱われており，位置づけが低いのが現状である．これは，FDT がまだ成熟していないうえに，専門家が十分でなく正しい形式記述仕様の作成・維持管理が困難であることに起因する．

今後，複雑化する一方のサービス／プロトコル仕様を厳密に記述し，維持管理していくためには FDT の重要性がより認識される必要があると考えられる．このため，FDT のツール開発普及などを含め一層の研究が必要である．

参考文献

1) ISO/IEC JTC 1/N 145 : ISO/IEC Joint Technical Committee 1, JTC 1 Statement of Policy on Formal Description Techniques (Dec. 1987).
2) Van Eijk, P., Vissers, C. and Diaz, M. ed. : The Formal Description Technique LOTOS, North-Holland (1989).
3) ISO/IEC 10026-2 : Information Processing Systems—Open Systems Interconnection—Part 2 : Transaction Processing Service Definition.
4) ISO/IEC DP 10026-3 : Information Processing Systems—Open Systems Interconnection—Part3 : Transaction Processing Protocol Specification.
5) ISO/IEC DTR 9571 : Information Processing Systems—Open Systems Interconnection—LOTOS Description of the Session Service—Type 2 (1988).
6) ISO/IEC DTR 9572 : Information Processing Systems—Open Systems Interconnection—LOTOS Description of the Session Protocol—Type 2 (1988).
7) ISO/IEC ISO 8802-3 : Information Processing Systems—Local Area Networks—Part 3 : Carrier Sense Multiple Access with Collision Detection Access Method (CSMA/CD) and Physical Layer Specifications.
8) 田畑他：OSI—明日へのコンピュータネットワーク，日本規格協会 (1987).
9) 棟上編：OSI の応用，日本規格協会 (1987).
10) 大特集：分散処理技術，情報処理，Vol. 28, No. 4 (Apr. 1987).
11) 大特集：ネットワークアーキテクチャ（開放型システム間相互接続）の標準化動向，情報処理，Vol. 26, No. 4 (Apr. 1985).

(平成元年 11 月 1 日受付)