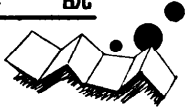


## 解説



## 開放型システム間接続 (OSI) の標準化†

元岡 達†† 苗村 憲司†††

## 1. ま え が き

電子計算機の機種、規模、適用業務は多様化の一途をたどり、その設置台数も大幅に増大している。これらの計算機と通信回線経由で接続して情報の入出力手段および遠隔の計算機へのアクセス手段を提供するデータ通信端末の多様化と普及もまた急激である。更に、計算機間通信方式、公衆データ網、分散処理方式などの技術進歩により、複数のオンライン情報処理システム（データ通信システム）を相互接続して複合的な業務を実現する例も多く見られるようになった。

利用者が目的に応じて計算機や端末あるいはこれらを含むシステムを選択し、それらの間を自由に接続して業務処理を実現するためには、これらの装置またはシステムを相互接続するための方式を標準化する必要がある。ISO（国際標準化機構）の TC 97（電子計算機と情報処理）はこの方式を開放型システム間接続（Open Systems Interconnection, 以下では OSI と略す）と呼び、その検討を目的とした SC 16 を設置して標準化活動に着手している<sup>1)</sup>。

本稿では、ISO/TC 97/SC 16 における検討状況を中心として、OSI の標準化動向<sup>2)</sup>、その参照モデル<sup>3)</sup>の概要などを紹介する。

## 2. 経 緯

計算機相互間の情報交換に関する ISO/TC 97 の標準化活動の歴史は長く、SC 2 におけるキャラクタコードの標準化、SC 6 における HDLC（ハイレベルデータリンク制御）手順の標準化<sup>4)</sup>などのように具体的な成果を生んできた。また、CCITT（国際電信電話諮問委員会）の SG VII（第7研究委員会）においても、公

衆パケット交換網のための勧告 X. 25<sup>5)</sup>、回線交換網のための勧告 X. 21 を代表とする各種の標準が作成されてきた。

しかし、近年の技術進歩により、更に次のような高位のプロトコル（通信規約）を定める必要が生じてきた<sup>6)</sup>。

(1) 計算機上に複数共存する利用者の業務処理プロセス同士を通信回線経由で接続してメッセージ転送を行うためのプロトコル。

(2) 多様な端末の属性を標準的な論理モデル（仮想端末）に変換し、計算機上の業務処理プロセスとの間ではこの論理モデルとして通信を行う仮想端末（virtual terminal）プロトコル。

(3) 計算機間でファイルを共用するためのファイルアクセス・転送プロトコル。

(4) ジョブを遠隔の計算機で実行するためのジョブ転送プロトコル。

これらのプロトコルを体系的に定めるためには、まずその環境を明確化する必要がある。「開放型システム間接続 (OSI)」とは、既存の各種標準と上述の高位プロトコルを合わせて全体を体系化するための環境を意味する新語だと言えよう。

ISO/TC 97 は、1977年3月のシドニー総会において、OSI を主題とする SC 16 の設置を決定した。SC 16 は、まず OSI の枠組を明確化するための参照モデル (Reference Model) を作成することとし、1978年2月～3月の第1回会議（ワシントン）、1979年6月の第2回会議（ロンドン）およびその間に開かれた数回の WG 会議の審議を経て、OSI の参照モデルを作成した。本モデルは後述（4.1 節）するように OSI の各種標準を開発するための共通基礎となるべきものであり、今後、改良を経て確定される予定である。

第2回 SC 16 会議では同時に四つの WG を編成して参照モデルの改良と各種の高位プロトコルの標準化を進める体制を整えた。各 WG は 1979年11月の TC 97 総会（マドリッド）の承認を得た後に本格

† Standardization for Open Systems Interconnection by Tohru MOTO-OKA (Faculty of Engineering, University of Tokyo) and Kenji NAEMURA (Yokosuka Electrical Communication Laboratory, N. T. T.).

†† 東京大学工学部

††† 日本電信電話公社 横須賀電気通信研究所

的な標準化検討に着手し、1980年秋頃の第3回SC16会議に結果を報告することとなっている。

本学会規格委員会の下にも1978年3月にSC16専門委員会が設置され、国内委員会としての精力的な活動を行った。更に国際会議にも積極的に参加し、わが国で開発されたデータ通信網アーキテクチャ (DCN-A)<sup>2)</sup>の階層モデルを紹介するなど、具体例を提示することによってISO側の検討に対してわが国の技術成果の反映を進めてきた。今後も更に具体的プロトコルの提案を行うべく検討を続けている。

### 3. OSIとネットワークアーキテクチャ

OSIとは何かについて以上にも簡単に述べたが、本章ではその概念をより明確に定義するため、まずネットワークアーキテクチャの基礎概念を紹介した後にOSIとの関連を述べることにしよう。

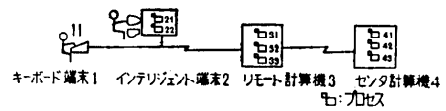
#### 3.1 ネットワークアーキテクチャの基礎概念

内外で多数のネットワークアーキテクチャが開発されているが、これらに共通的な基礎概念は通信機能の階層化である<sup>2),6),7)</sup>。

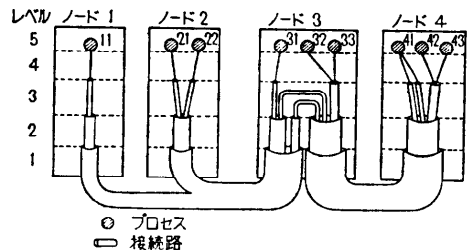
図-1(a)の簡単なデータ通信系を例にとってこの基礎概念を説明しよう。同図では、計算機3、4およびインテリジェント端末2の中で走行中の業務処理プロセスが相互に通信を行っているほか、左端のキーボード端末1のオペレータ11がセンタ計算機4のプロセス41と通信をしている。リモート計算機3は端末1、2とセンタ計算機4との間で中継の役割も果たしている。

ネットワークアーキテクチャでは、オペレータの操作をも計算機やインテリジェント端末内のプログラムによる処理と同等にとらえ、ネットワークの構成装置間の通信をすべて「プロセス間通信」として取扱うのが普通である。更に、各装置の通信機能を統一するため、各装置を「ノード」という論理モデルに対応させその中を複数の階層に分割することが多い。

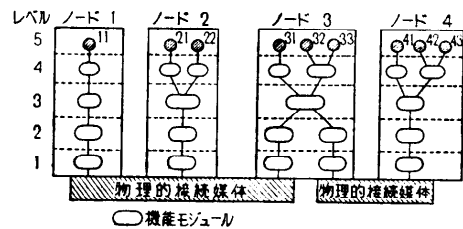
この観点から図-1(a)をモデル化すると同図(b)、(c)が得られる。同図(b)では通信のための接続路の階層構造を表現しているのに対し、同図(c)ではプロトコルを実行する機能モジュールの結合関係を表現している。このモデルでは、各ノードの機能を①物理接続制御(例えばモデム制御)、②データリンク制御(例えばHDLC手順)、③トランスポート制御(通信相手のノードに至る通信接続路の制御)、④機能制御(プロセス相互間の通信の制御)および⑥情報



(a) 簡単なデータ通信系の構成例  
(ここで、次の組合せの通信(プロセス間通信)が行われているものとする: 11-41, 21-41, 22-31, 32-42, 33-43)



(b) 通信機能の階層化モデルの図式表現 (1)  
(通信のための接続路に着目した表現)



(c) 通信機能の階層化モデルの図式表現 (2)  
(プロトコルを実行する機能モジュールに着目した表現)

図-1 データ通信系の階層化モデルによる表現例

処理(プロセス)の五つのレベルに分割している。

一般にネットワークアーキテクチャでは、これと類似の機能分割を行い、更に詳細な分割を行った上で個々のプロトコルを定めている例が多い。また、原則として実際のハードウェア、ソフトウェアの機能モジュールを階層化モデルに対応付けて設計することが多い。

#### 3.2 OSIの概念

OSIの思想は、根本的にはネットワークアーキテクチャの思想と軌を一にするものであるが、特に次の2点に特徴がある。

(a) 利用者が、その目的に応じて計算機や端末等を選択し、それらの間を自由に接続して業務処理を実現するために必要な各種の標準を全体的に体系化することがOSIの最終的な目標である。このためには利用者向きプロトコルを含め、すべてのプロトコルを標準化の対象として位置付ける大掛りな論理モデルを設定する必要がある。

(b) 一方、すべての装置間で(a)に述べた標準

化プロトコルを適用することは必ずしも効率的でない場合がある。例えば同一室内に設置した同一メーカー製品の間では非標準 (OSI に従わない) のローカルなプロトコルを使用することを認めてもよい。

この観点から、1台ないし複数台の装置 (計算機、端末等) とそのソフトウェア、オペレータなどを組合せた1式のシステムを単位と考え、このシステムが、他のシステムと通信する場合に OSI 標準に従う場合にこのシステムは、open system (以下では簡単のため単に system と書く) であるとみなす。system 内の装置間の通信は必ずしも OSI 標準に従わなくともよい。OSI とは、このような system を通信回線網などの物理媒体に接続し、system 上の application-process (利用者プロセス、管理プロセス、オペレータ等) の間で通信を行い、分散処理を実行するために守るべき諸標準と、これらの標準の総合的相互関連を記述した文書 (参照モデル) との体系であると定義できる (図-2 参照)。

参照モデルにおいては、ネットワークアーキテクチャの場合と同じように system を複数の階層に分割するが、上述の理由からその分割は必ずしも実際のハードウェア、ソフトウェアの機能分担とは対応しないでもよいことになり、system 内構成の自由度が保たれる。

#### 4. OSI の参照モデル

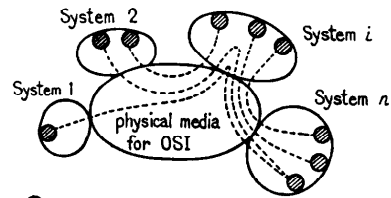
本章では、ISO/TC 97/SC 16 資料 N 227<sup>3)</sup> に示された参照モデルの概要について述べる。なお同資料は検討途中の "Working document" であり、今後検討すべき項目 (further study 項目、以下では FS と略す) を多数含むほか、記述された内容にも今後変更が予想されるので注意されたい。

##### 4.1 参照モデルのねらいと適用対象

###### (1) OSI 参照モデルのねらい

OSI 参照モデルは、OSI のための各種標準の開発作業を相互調整するための共通基盤を提供し、更に既存の標準を全体の参照モデル内に位置付けるための便宜を与えることをねらいとしている。また、新たに標準を開発するか既存の標準を改良すべき分野を識別すること、関連する諸標準の間の無矛盾性を維持するための参考となることも本資料のねらいに含まれる。

しかし、このモデルは、実際のハードウェア、ソフトウェアの実現方法についての仕様を定めるものではなく、標準を定めるための概念的、機能的な枠組を設



● application-process  
---- connection

図-2 OSI による Open System 間の接続

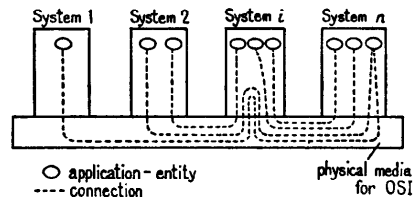


図-3 N 227 参照モデルの基礎構造

定するものである。

###### (2) 資料 N 227 の適用対象に関する制約

OSI 参照モデルは広範な対象を記述すべきものであるが、現在までに作成された N 227 では記述対象に制限を付している。特に重要な制約は次の項目である。

(a) 相互接続の物理媒体としては電気通信回線を用いるものとする。磁気テープ輸送など、他の媒体による場合は FS とする。

(b) 主に通信 (プロセス間通信) の問題を取扱う。プロセス管理など、他の問題は FS とする。

(c) 主にコネクション型の通信を対象とする。トランザクション型、ブロードキャスト型などの扱いは FS とする。

###### (3) N 227 参照モデルの基礎構造

前項の制約の下で N 227 が記述する参照モデルの基礎構造は図-3 の通りである。即ち、本モデルは次の三つの要素で構成される。

(a) 物理媒体 (としての電気通信回線) に直接接続された複数の system。

(b) 各 system 上に存在する application-entity。ここでいう application-entity とは、3章に述べた application-process の一部分であって、OSI 環境下の通信に関与する機能をモデル化したものをいう。

(c) 異なる system 上の application-entity 相互間を結ぶ connection。

#### 4.2 階層化構造の一般概念

N 227 において参照モデルを構成する最も重要な思想は通信関連機能を複数の layer (層) に分割する

layering の概念である。本節では、layering の基本概念、これに基づいて定義される (N)-connection, (N)-protocol-data-unit などの一般概念について述べる。

(1) layering の基本要素

本モデルで対象とする (通信関連) 機能の全体を複数の layer に分割する。分割の原則、並びに各 layer に与えられた役割については後述 (4.3 節) する。ここでは各 layer に順序が付されていることだけに注意し、物理媒体に近い方 (lowest) から順に 1, 2, ..., N-1, N, N+1, ... の layer 番号を与える。簡単のため番号 N の layer を (N)-layer と呼び、(N)-layer において定義される各種の概念に (N)- を付して識別する。

全体を layer に分割することにより、各 system も layer 対応の subsystem に分割される。各 subsystem には 1 個ないし複数個の entity が存在する (entity は、図-1 (c) の機能モジュールをモデル化した概念と考えてよい)。(N)-layer の entity 即ち (N)-entity は、直下の (N-1)-layer から提供されるインタフェース機能即ち (N-1)-service を利用し、更に他の (N)-entity と協力して自 layer 内機能即ち (N)-function を実行することにより、直上の (N+1)-layer に提供すべきインタフェース機能即ち (N)-service を実現する。この目的を達成するために、複数の (N)-entity の間で情報を送受 (通信) するための規約と形式の体系を (N)-protocol と呼ぶ (以上、図-4 参照)。

各 (N)-entity が直下の (N-1)-layer から (N-1)-service の提供を受けるためのアクセス点を (N-1)-service-access-point と呼ぶ。同様に、(N)-entity が直上の (N+1)-layer に対して (N)-service を提供するアクセス点は (N)-service-access-point である (図-5 (a) 参照)。

(2) 各種の connection

4.1 節では application-entity (即ち最上位 layer の entity) を相互に結ぶ通信接続路として connection を導入したが、ここではこの概念を拡張し、各 (N)-layer の提供する (N)-service の一種として (N)-connection を定義する。

(N)-connection とは、複数の (N+1)-entity の間の結合関係 (association) であり、各 (N+1)-entity につながる (N)-service-access-point を相互に結ぶ通信路として (N)-layer により提供される。(N+1)-

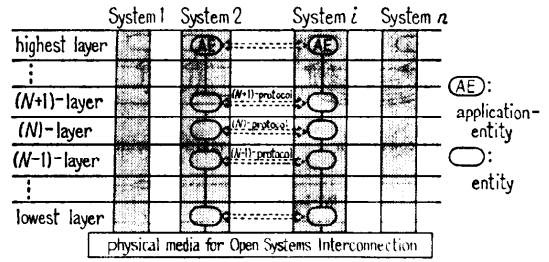
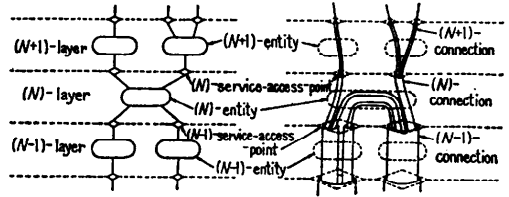


図-4 layering の構造



(a) layering の基本要素 (b) connection との対応

図-5 layering の基本要素と connection との対応

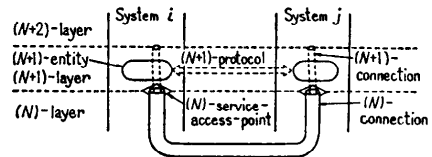


図-6 connection と entity と protocol の関係 ((N)-connection によって結ばれた二つの (N+1)-entity が (N+1)-protocol に従って通信する)

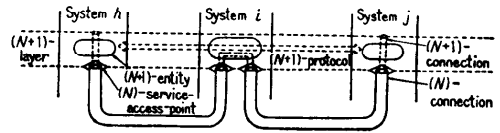


図-7 relay による通信

(system i の (N+1)-entity が relay を行う。)

entity はこれを利用することにより (N+1)-protocol に従った通信を行い、これによって新たに (N+1)-connection を実現して (N+2)-layer に対して提供する (図-6 参照)。

なお、1本の (N)-connection で直接結合されていない (N+1)-entity 同士が通信を行うために他の (N+1)-entity の relay (中継) 機構を利用する場合は、見掛け上、二つの (N)-connection が直列に連結されたような形になり、これを通して連続した (N+1)-connection が実現される (図-7 参照)。このような relay 機構を応用して複数の (N)-connection を直列に通した (N+1)-connection を実現する (N+1)-function を routing と呼ぶ。

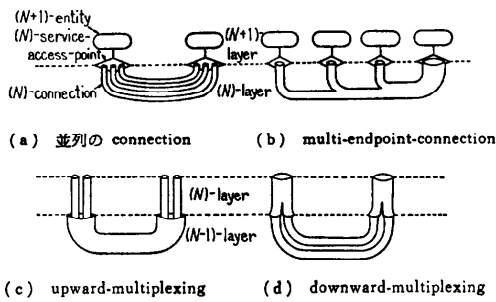


図-8 (N)-connection に関する諸特性

以上のほか、(N)-connection の特性として重要な項目には次のものがある(図-8 参照)。

(a) 1対の(N+1)-entity の間に複数の(N)-connection を並列に設定すること(用途によって使いわけることが考えられる)。

(b) 3個以上の(N+1)-entity の間に1本の(N)-connection を設定すること(これを multi-endpoint-connection と呼ぶ。図-1 (a)(b) の分岐回線を一般化したものと考えられる)。

(c) 1本の(N-1)-connection を利用して複数の(N)-connection を実現すること(upward-multiplexing。下位の connection の有効利用のため必要)。

(d) 複数の並列(N-1)-connection を利用して1本の(N)-connection を実現すること(downward-multiplexing。下位の connection の障害対策などに用いられる)。

以上の特性を含めた connection の階層構成例を図-5 (b) に示す。

また、(N)-connection を通して二つの(N+1)-entity 間で送受するデータの方向としては次の3通りの場合がある。

(i) one-way (一方向のみにデータを転送する)。

(ii) two-way-alternate (両方向で交互にデータ転送が可能)。

(iii) two-way-simultaneous (両方向で同時にデータ転送が可能)。

(3) 各種の identifier

一般に各 layer には多数の entity が存在し、その直下の layer から提供される connection も多数になる。通信を行うに当たっては特定の entity を相手として指定し、特定の connection を使用する必要がある。このため、個々の entity や connection を識別する名前やアドレスに相当する identifier が次のように定義されている。

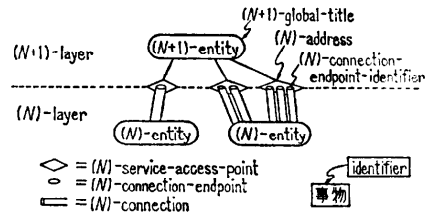


図-9 entity, service-access-point, connection-endpoint および各々の identifier

(a) 個々の(N)-entity を(N)-layer 内で一意に識別するための(N)-global-title。

(b) 個々の(N)-service-access-point を識別するための(N)-service-access-point-address (簡単のため、以下では単に(N)-address と呼ぶ)。

(c) 個々の(N)-connection をその端(endpoint)において識別するための(N)-connection-endpoint-identifier。

図-9 に示すように一つの(N+1)-global-title に対して1ないし複数の(N)-address が対応し、一つの(N)-address に対して1ないし複数の(N)-connection-endpoint-identifier が対応する。

これらの概念を用いて connection の使い方を簡単に説明しよう。

ある(N+1)-entity が他の(N+1)-entity との間に(N)-connection を必要とする場合は、まず相手の(N+1)-global-title を指定し、これに対応する一つの(N)-address を求める。ここで、(N+1)-global-title から(N)-address を求める機能は(N+1)-function の一種であり、(N+1)-directory と呼ばれる(例えば、通信管理プログラムがネットワーク構成制御機能の一環として維持するアドレス対応表として実現されよう)。

次に、この(N+1)-entity は自分の直下の(N)-entity に対して相手(N+1)-entity の(N)-address を示し、(N)-connection の設定を要求する。(N)-entity は要求された(N)-connection を設定するとその端に(N)-connection-endpoint-identifier を付して(N+1)-entity に通知する。

このあと、(N+1)-entity がデータの送受を行うに当たっては、相手の(N+1)-entity と両者間の(N)-connection とを指定する代わりに(N)-connection-endpoint-identifier のみを指示すればよい(この考え方は CCITT 勧告 X. 25 パケットレベルプロトコルの論理チャンネル番号<sup>5)</sup>に類似しており、並列 connec-

tion の利用がしやすいことなどの利点がある)。

(4) 各種の data-unit

以上に述べたことから分かるように、二つの system 間でデータを送受するためには、各 layer の entity がそれぞれ connection を利用し、protocol に従って通信を行う必要がある。connection 上を流れる情報の転送単位は data-unit と呼ばれ、種々の観点から定義されているが、特に重要な data-unit は次の 2 種である。

(a) (N)-service-data-unit とは、(N)-connection の両端でひとまとまりの情報として受け渡される単位を言う。

(b) (N)-protocol-data-unit とは、(N)-entity 相互間で (N)-protocol に従って送受信される情報の単位である。その内容は、(N)-protocol の実行のために必要な (N)-protocol-control-information (通常はヘッダとして先頭に付けられる) と、(N)-service-data-unit 又はその部分に相当する (N)-user-data (一般にテキスト又は情報フィールドと呼ばれている部分) とに分けられる。

最も簡単な場合 (図-10) には、(N)-service-data-unit がそのまま (N)-user-data と 1 対 1 に対応し、更に (N)-protocol-data-unit がそのまま (N-1)-service-data-unit と 1 対 1 に対応する。この場合には送信側 system では layer ごとに user-data に protocol-control-information (ヘッダ) を追加し受信側 system では layer ごとにヘッダを削除する処理を行うこととなる。

より一般的には、layer ごとに connection の転送能力などに差があるため、data-unit の対応が 1 対 1 に限定されず、図-11 に示すように segmenting 又は blocking が行われることが多い。

また、data-unit の転送に際しては flow control(流量制御)、sequencing (順序制御)、error control (誤り制御) などの機能が service および function として定義される。更に、通常の data-unit の流れとは別に、これよりも高い優先順で転送され、割込みなどの目的に応用可能な expedited-data-unit の概念も定義されている。

4.3 7-layer model の設定

4.1 節 (2) に述べた制約の下で 4.2 節の一般概念を適用するために N 227 で使用した layer 分割の原則、並びにこの原則に基づいて選ばれた 7-layer model における各 layer の役割の概要は次の通りである。

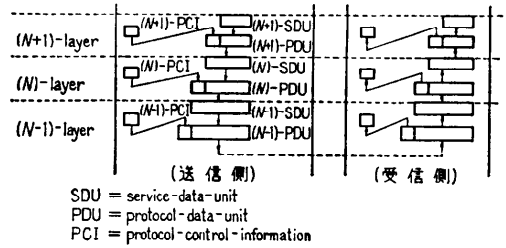
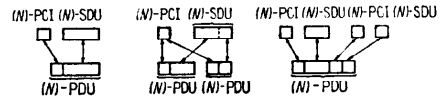


図-10 service-data-unit と protocol-data-unit の関係 (one-to-one mapping の場合)



(a) one-to-one mapping (b) segmenting (c) blocking

図-11 (N)-service-data-unit と (N)-protocol-data-unit の関係

(1) layer 分割の原則

特定の layer 分割が最適であることを証明するのは困難であろうが、分割方法に関するいくつかの原則を定めることができる。N 227 では、次の例に示すような原則を 13 項目示している。

- (a) 隣接 layer 間の境界で定義すべき service の記述がなるべく簡単になるように境界の場所を選ぶ。
- (b) 特に他と質の異なる function を集めて一つの layer とする。
- (c) 特定の layer 内の function や protocol を完全に入れ換えても他の layer に影響がないように分割する。

(2) 7-layer model の大要

上の原則に基づいて設定された N 227 参照モデルでは、最上位から順に Application, Presentation, Session, Transport, Network, Data Link, Physical の七つの layer を定義している。各 layer には表-1 に示すように役割を与えている。

4.4 各 layer の機能概要

本節では、N 227 の 7-layer model における各 layer の機能 (直上 layer に提供する service および自 layer 内の function) の概要を述べる。なお、4.2 節に示した一般概念との対応を表-2 に示す。

(1) Application Layer

最上位の layer であり、application-entity が存在してこの間で application-protocol を実行する。元来 application-entity は application-process の中に含まれる OSI 向き通信関連機能を抽出してモデル化した

表-1 各 Layer の役割

N	layer の名称	役割
7	Application Layer	Application-entity の存在する最上位 layer であり、管理用および利用者向けの application-protocol を実行する。
6	Presentation Layer	構造を持つデータの入力・授受・表示・制御を行う。仮想端末プロトコル、および仮想ファイルプロトコルとジョブ転送プロトコルの基本部分を実行する。
5	Session Layer	session と呼ぶ connection を設定し、データの送信・受信の制御、同期の制御などを行う。
4	Transport Layer	session-entity の間に汎用的な transport-connection を設定し、トランスペアレントなデータ転送を行う。
3	Network Layer	一つ又は複数の通信網を通して routing を行い、通信相手の application-entity の存在する system に至る network-connection を提供する。
2	Data Link Layer	二つの system の間でデータを送受信し、物理的通信回線の伝送誤りに対処する data-link 制御手順を実行する。
1	Physical Layer	physical-connection (即ちデータ回線) を設定・維持・解除するための機械的・電氣的・機能的および手順上の規約を含む。

ものであり、次の application-process の種別に応じて表-3 のように各種の protocol を実行する。

- (a) system-management-application-process (system の共通資源を管理する application-process)
- (b) application-management-application-process

表-3 Application-process の分類と application-protocol の種別

application-process の分類	application-protocol の種別と機能
(a) system-management-application-process	(i) system management protocols(資源の活性化・非活性化, モニタ機能, 誤り制御など)
(b) application-management-application-process	(ii) application management protocols (アクセス権制御, 課金, インテグリティ制御, セキュリティ制御など)
(c) user-application-process	(iii) system protocols (ファイルアクセス, ジョブ入力など) (iv) industry-specific application-protocols (金融業用など) (v) enterprise-specific application-protocols (特定企業の受入力用など)

(利用者の業務処理に対応した資源へのアクセス権の管理などを行う application-process)

(c) user-application-process (利用者の業務に対応した情報処理を行う application-process)

(2) Presentation Layer

端末の入出力情報、ファイル、ジョブ入出力情報などのように構造を持つデータの入力、授受、表示、制御を行う layer である。

直上の Application Layer に対して提供する最も重要な service として、presentation-image と呼ばれる

表-2 一般概念の各 layer への適用

N	7	6	5	4	3	2	1	
(N)-layer	Application Layer	Presentation Layer	Session Layer	Transport Layer	Network Layer	Data Link Layer	Physical Layer	
(N)-entity	application-entity	presentation-entity	session-entity	transport-entity	network-entity	data-link-entity	physical-entity	
(N)-connection	—	(FS)	session	transport-connection	network-connection	data-link-connection	physical-connection	
(N)-connection に関する諸特性	並列 (N)-connection	—	(FS)	○	○	○	○	(FS)
	multi-endpoint	—	(FS)	(FS)	—	(FS)	(FS)	○
	upward-multiplexing	—	—	(FS)	○	○	—	—
	downward-multiplexing	—	—	—	(FS)	(FS)	(FS)	—
	one-way	—	(FS)	○	—	—	(FS)	—
	two-way-alternate	—	(FS)	○	—	○	○	○
two-gway-simultaneous	—	(FS)	○	○	○	○	○	
routing	—	—	—	—	○	—	(FS)	
segmenting, blocking	—	—	(FS)	○	○	(FS)	(FS)	
flow control	—	—	(FS)	○	(FS)	○	—	
sequencing	—	—	○	○	(FS)	○	○	
expedited-data-unit	—	—	○	(FS)	(FS)	(FS)	—	
protocol の例	管理用プロトコル, 利用者向きプロトコル (各 1 種)	仮想端末プロトコル	セッションプロトコル	トランスポートプロトコル	CCITT 勧告 X.25 パケットレベルプロトコル	HDLC, CCITT 勧告 X.25 フレームレベルプロトコル	CCITT 勧告 X.21	
		仮想ファイルプロトコル ジョブ転送プロトコル						

備考: ○=有り又は可能, —=無し又は不可能, FS=further study

表-4 仮想端末, 仮想ファイル, ジョブ転送の機能の割当て

区分	layer	機能大要
仮想端末 プロトコル	Presentation Layer	(1) 仮想端末プロトコルの初期化 (a) 端末クラスの選択 (ストリーム・クラス/ページ・クラス, 他) (b) プロファイルのネゴシエーション (例えば行内桁数の指定) (2) データの授受 (例) data-store のアドレッシング, 書き込み, 表示, データ伝送の開始指示, 割り込み指示, 書式の記述と割当て (3) 操作の制御 data-store へのアクセス状態に応じた制御
	Application Layer	(1) Filestore (一つの system 内のファイルの集合) 関連の手続き (例) 資源割当て制御, integrity 保証のための commitment 制御 (2) ファイル関連の手続き (a) ファイルの生成 (b) ファイル・アクセス (オープン/クローズ, ファイル内位置決め, レコードの読出し/書き込み/削除/ロック) (c) ファイル転送 (複写/転送後原ファイル削除) (d) ファイルの削除 (e) ファイルの連結
仮想ファイル プロトコル	Presentation Layer	(1) 外部属性の記述 (例) ファイル名, ファイル構造, アクセス制御, データ保護, 課金の可否, 履歴, 要求品質, integrity 制御 (2) 内部属性の記述 (例) サイズ, コード表現 (2進/キヤラクタ・コード/外部指定/文書用など), データ型
	Application Layer	(例) ジョブの転送/出力転送/処理状態の通知/キャンセル, オペレータ間通信, システムメッセージ転送, ユーザファイル転送
ジョブ転送お よび取扱いプ ロトコル	Presentation Layer	(例) RJE 向きデータ表現, レコード構造とデバイス制御, 機能の表現, 仮想ジョブ・仮想ホスト管理/アクセス用コマンドの表現

表-5 session-service の大要

分類	項目	説明
(a) session-administration-service	session の設定・解除	1本の transport-connection に対して1本の session を設定する。但し, 時間的に重ならないければ複数の session が1本の transport-connection を順に使ってもよい。
	コンテキスト管理	application-protocol および presentation-protocol の識別, 切替え
	session 識別	identifier の提供
	session 回復	session 障害時の正常動作の再開
(b) session-dialogue-service	データの区切り	(i) session-service-data-unit (SSDU). (ii) quarantine-unit (1個又は複数個の SSDU の系列で, 全体を受信して初めて意味のあるもの, QU). (iii) session-interaction-unit (1個又は複数個の QU の系列で, 連続して一方方向に送られるもの, SIU). (iv) commitment-unit (1個又は複数個の SIU の系列で, データのインテグリティ制御の単位となるもの, CU) (v) recovery-unit (1個又は複数個の SSDU の系列で CU に含まれ, 誤り回復の単位となるもの)。
	データ転送方向の管理	(i) one-way (ファイル転送などに用いる) (ii) two-way-alternate (端末送信などに用いる) (iii) two-way-simultaneous

論理モデルの制御機能がある。presentation-image とは, session を通して見えるデータ構造と相手の presentation-entity の動作をモデル化したものであり, 仮想端末用, 仮想ファイル用, ジョブ転送用などの presentation-image が定義される予定である (但し, その具体的内容は未定であり, Presentation Layer と Application Layer の境界を明確化することが FS となっている)。

代表的 protocol に対応する機能例を次に示す (表-4 参照)。

(a) 仮想端末プロトコル——端末クラスの選択, プロファイルのネゴシエーション, データとコマンドの転送, 書式管理

(b) 仮想ファイルプロトコル——仮想ファイルストアコマンドの形式制御, ファイルデータとコマンドの通信, コード変換

(c) ジョブ転送プロトコル——レコード構造とデ

バイスの制御, コマンドの形式制御, データの形式制御  
なお, このほかに layer 内の function としてデータ圧縮および暗号化がある。

(3) Session Layer

本 layer の主目的は, 通信する application-entity の直下の presentation-entity 同士を結ぶ connection (実質的には図-2, 図-3 に示した connection に相当する)として session を提供することである。session-service は次の2種に分類される (表-5 参照)。

(a) session-administration-service (session の設定・解除, コンテキスト管理, session 識別, session 回復)

(b) session-dialogue-service (データの区切り, データ転送方向の管理)

(4) Transport Layer

本 layer は session-entity の間に汎用的な transport-connection を設定し, トランスペアレントなデータ転送を行うことである。



transport-connection の特性は 4.2 節の一般概念に示した通りである(但し、表-2 のように制限がある)。その位置付けについては、実質的に図-2、図-3 の connection に相当するという見方と、system 間の connection であるという見方とがあり FS となっている。

#### (5) Network Layer

一つ又は複数の通信網を通して routing を行い、通信相手の application-entity の存在する system (内の transport-entity) に至る network-connection を提供する layer である。

network-connection の特性は、4.2 節の一般概念に示した通りである(但し表-2 の制限がある)。

network-protocol の例としては CCITT 勧告 X.25 のパケットレベルプロトコルがある。なお、公衆パケット網のほかにも専用線等をつないで構成される通信網を用いて両端の system 間を結ぶ network-connection を設定するためのプロトコルもこれに含まれる。但し、network-connection 上のデータ送達確認機能を network-service に含めるか否かは FS となっている。

#### (6) Data Link Layer

通信回線で直結された二つの system(内の network-entity) 間に data-link-connection を提供し、この上でデータ伝送誤り制御等を実行する layer である。

代表的な data-link-protocol として HDLC がある。CCITT 勧告 X.25 フレームレベルプロトコルもこれに該当する。

#### (7) Physical Layer

複数の system (内の data-link-entity) 間に physical-connection 即ちデータ回線を設定し、維持し解除するための機械的、電気的、機能のおよび手順上の規約を含む。データ回線には直通回線、交換回線、分岐回線 (multi-endpoint-physical-connection) があり、これらの制御手順が含まれる。

代表的なプロトコルの例として CCITT 勧告 X.21 などがある。

### 4.5 残された課題

4章の最初に断ったように、N 227 参照モデルは検討途中の資料である。4.1 節(2)や4.4節内で述べたように FS とされている課題も多い。現に N 227 自身の中に多数(約 80 件)の FS 項目が明示されており、この中には、上述の項目のほかに例えば次のような課題が含まれている。

#### (1) Sublayer の導入

4.3 節に述べた layer 分割の原則の一環として、

layer 内を更に複数の sublayer に分割する可能性が示唆されている。sublayer は、オプション機能の分別、将来の機能追加に対する柔軟性などの点から導入が提案されており、具体例として Network Layer を複数の sublayer に分割することなどが FS となっている。

#### (2) 非標準プロトコルのサポート

既存システムが段階的に OSI 標準を導入することを可能とするため、特定の layer の標準プロトコルの代りに非標準プロトコルを用いるための escape 機構が FS となっている。

(3) データのインテグリティ、セキュリティ、保護など管理プロトコルとの関係も含めて FS となっている。

## 5. むすび

ISO/TC 97/SC 16 を中心とした開放型システム間接続(OSI)の標準化動向と OSI の参照モデルの概要について紹介した。

具体的なプロトコルの標準化は今後 SC 16, SC 6 などにおいて本格化する予定である。本学会規格委員会の専門委員会においても熱心な検討が行われている。

一方、CCITT でも SG VII の下で layered model の検討が本格化しており、上述の参照モデルに類似のモデルが審議されている。また、ISO と CCITT の間で分担を明らかにした協力関係を結ぶことが提案されており、今後、両者の協調の下で統一された国際標準が作成されることが望まれる。

## 参考文献

- 1) 規格委員会: 1978 年における規格委員会の活動, 情報処理, Vol. 20, No. 6, pp. 527-544(1979).
- 2) 信国: 国際標準化活動の動向, 情報処理, Vol. 20, No. 6, pp. 545-550 (1979).
- 3) ISO/TC 97/SC 16 N 227: Reference Model of Open Systems Interconnection (Version 4 as of June 1979).
- 4) 川端, 田中: ハイレベルデータリンク制御手順とその標準化動向, 情報処理, Vol. 19, No. 5, pp. 456-464 (1978).
- 5) 石野: パケット交換網の通信規約, 情報処理, Vol. 18, No. 11, pp. 1148-1156 (1977).
- 6) 田中: プロトコル(通信規約), 信学会誌, Vol. 61, No. 10, pp. 1105-1113 (1978).
- 7) 戸田, 中田: データ通信網アーキテクチャ(DC-NA)の基本概念, 情報処理, Vol. 20, No. 2, pp. 153-158 (1979).

(昭和 54 年 10 月 5 日受付)