

解 説

4. 上位層のサービスとプロトコル



4.5 仮 想 端 末†

松 下 温†† 相 田 潔††

1. は じ め に^{1),2)}

遠隔地におかれた端末がコンピュータと通信する場合、初期にはコンピュータが端末を直接制御した。すなわち、端末ごとの物理的な特徴（印字幅、ハードウェア制御命令、動作時間等）に合わせて個々の端末を動作させる必要があった。これではコンピュータ側が実在する端末の種類に応じて端末制御用のソフトウェア等を多数もつ必要があり、好ましくないことは明らかである。そこで、端末制御をセンタのコンピュータの代わりに実行する端末制御装置が登場した。これにより、センタのコンピュータは個々の端末の物理的な特徴が異なることによる複雑さからある程度解放された。

コンピュータシステムの利用形態に対する要求は分散処理の浸透、公衆データ網利用による異機種コンピュータの相互接続など、次々に拡大してきている。さらに、同一端末から異機種複数のコンピュータへの接続が要求されるようになってきた。このような状況になると、端末制御装置の導入だけでは不十分であることが露呈される結果となった。すなわち、ホストコンピュータから個々の端末に依存した制御を可能な限り解放するとともに、端末からは異なるホストコンピュータへ同一の方法でアクセスできるようにすることが必要となってきた。

仮想端末の概念はこの目的を達成するための手段として考え出されたものである。仮想端末とは、各種端末のもつ機能を抽象化して、各端末のもつ特殊機能を一般化し、標準化した仮想的な端末のことをいう。すなわち、ホストコンピュータおよび端末は互いに仮想端末と呼ばれる標準端末を介して通信することにより、ホスト側および端末側ともに多種類の通信を意識

する必要がなくなり通信対象がともに仮想端末に一本化される。この様子が図-1に示されている。図-1(a)はホストおよび端末での仮想端末の見え方を示している。ホスト上のユーザプロセスあるいは端末オペレータからは仮想端末が提供する機能が見える。一方、ネットワークの内部からは仮想端末の仕組み（機構）が見える。この機構は仮想端末プロトコルに従って動作する機構であり、言い換えれば、ホスト上にある端末制御部および端末側のホスト対応実端末制御部がともにもっている機構である。図-1(b)は仮想端末のネットワークアーキテクチャ上での位置付けを示している。アプリケーションまたは端末オペレータに見える仮想端末の機能はともに同じであるが、その機能の具体的な利用法は双方で異なってもよい。言い換えれば、仮想端末機能の具体的な利用法はローカルな問題であり、仮想端末が提供する機能そのものではない。

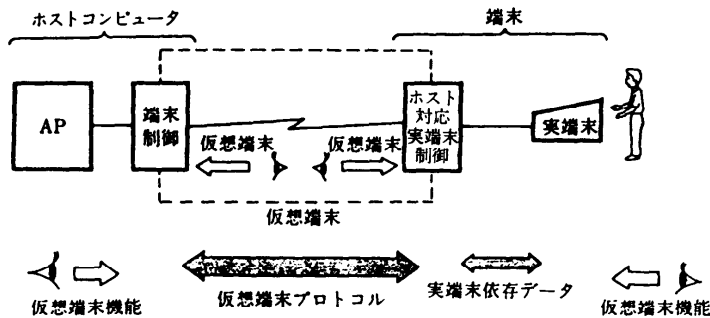
前述のごとく、仮想端末は種々の実端末のもつ機能を抽象化して表現することにより、コンピュータと端末間通信および端末相互の通信を可能な限り統一化するために導入された概念である。ここでいう種々の実端末のもつ機能の抽象化とは標準的な端末機能を設定することにほかならない。標準的な端末機能を抽出するためにはどのようにすればよいのであろうか。このためには端末として具備すべき機能の論理モデルを設定することが望ましい。

実端末のもつ種々の機能を表現できるような論理モデルを設定するために、種々の端末が現実にとどのような機能ブロックから構成されているかを考察する必要がある。図-2に端末の典型的な機能ブロック構成が示されている。図-2の(a)と(b)の差は制御部がハードウェア論理で構成されているか、あるいは制御部内にプロセッサをもちソフトウェア論理と合わせて機器を制御するように構成されているかによる。

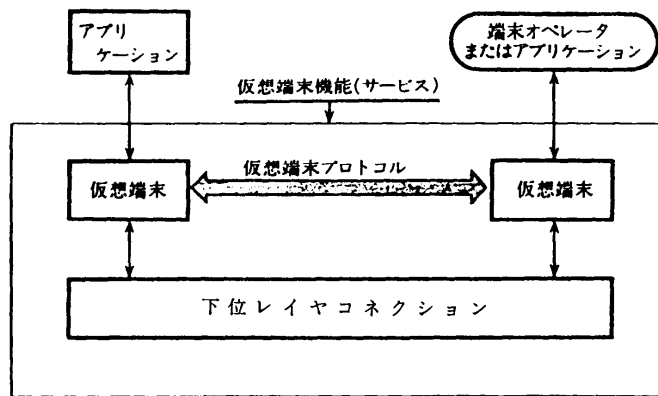
このように端末の機能ブロック構成は制御部の構成方法を除けばどのような端末でもほぼ同様であり、端

† Virtual Terminal by Yutaka MATSUSHITA and Kiyoshi AIDA (Computer Systems Division, OKI Electric Industry Co., Ltd.).

†† 沖電気工業(株)情報処理事業部システム本部



(a) 仮想端末の見え方

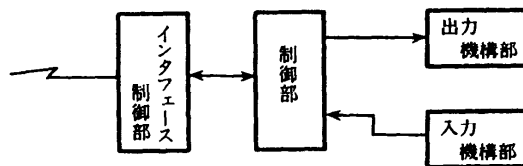


(b) ネットワークアーキテクチャにおける仮想端末

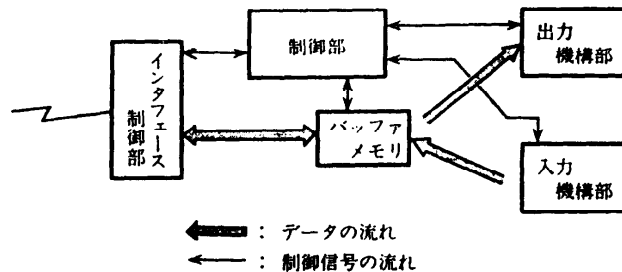
図-1 仮想端末の位置付け

末の機能はそのほとんどが制御部の機能により表わされる。したがって仮想端末の論理モデルの設定にあたっては、制御部の機能を適切に表現できることが必要である。今後はプロセッサ内蔵のインテリジェントの端末が主流になることが予想されるため、端末ソフトウェアあるいはファームウェアの機能を適切に表現できる論理モデルを設定することが重要となる。

ISO/TC 97/SC 16 の WG 5-VT (Virtual Terminal) グループはフランスの Naja Naffa が主査となり、イギリス、アメリカ、ドイツ、日本などが委員となって 1980 年 4 月より討論が開始された。1982 年 2 月に主査が同じフランスの Michael Lilly に交代するまでの間、検討の中心は論理モデルの開発であった。1981 年 6 月に図-3 に示す最新モデルの原型がほぼ固まっていた。

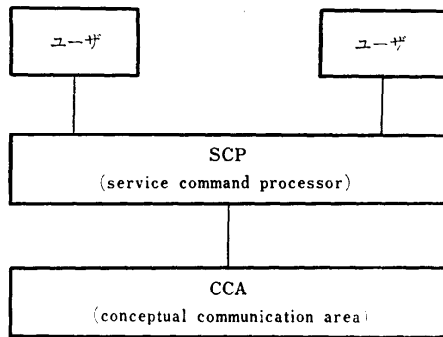


(a) 簡易(ノンインテリジェント)端末の機能ブロック構成



(b) インテリジェント端末の機能ブロック構成

図-2 端末の機能ブロック構成例



CCA: データ格納とそれに伴う制御をもつ4つの要素の集合。

SCP: ユーザとのサービスインタフェースを司る制御機構の集合。(a) モデルの構造

構成要素名	概 要	関連制御例
CDS (conceptual data store)	ユーザが業務で扱うデータを格納する。視覚化表示の対象である。	画面表示 用紙印字
CSS (control signalling and status store)	業務を遂行するうえで付随的に発生するデータを格納する。	ファンクション・キー 制御 ランプ点灯
DSD (data structure definition)	CDS, CSS, および AC を定義する情報を格納する。	—
AC (access control)	通信を実行する際に発生する動的な情報を記憶する。	—

(b) CCA の構成要素

制 御 機 構	概 要
配 送 制 御	CCA 内データのユーザへの出力 (ユーザによる読出し) を制御する。
アクセス制御	ユーザから CCA へのアクセスを制御する。
トリガ制御	関連の制御信号に連動させて上記制御機構を起動する。

(c) SCP の典型的な制御機構

図-3 仮想端末の一般論理モデル案

当時のモデルは図-2 の制御部の抽象化に相当する CCA (Conceptual Communication Area) の機構を図-3(b) の各部分に分解しただけに止まっていたが、論理モデルの最も重要な点であるデータ格納部を設けること (すなわち CDS と CSS) とデータ格納部に入れられるデータのタイプと構造化を可能にすることはすでに行われていた。

Lilly の時代になって、論理的なサービスおよびプロトコルを動作させるために必要な機構、OSI 参照モデル上の位置付け、具備すべき標準機能の過不足などの点から見直しが行われた。結果図-3 のモデルが得られた。図-3 で図-1(b) の二つの仮想端末が CCA という共通の箱で描かれていることは、次のような諸点を強調するためである。すなわち、双方のユーザに見える映像は同一であること、双方のユーザは CCA を共用していること、などの点である。

このような論理モデルは仮想端末の最大構成の機能の概要を表現することによって機能の枠組みを定めるために必要なものであり、構成要素の細部は現実の端末に則して定める必要がある。細部を定める際に仮想端末だからといってあらゆる種類の端末の機能をいたずらに統一すると、特定の端末から見た場合に機能の

冗長や機能の矛盾が発生する。これを避けるために仮想端末のクラス分けが行われる。クラス分けは端末の機能差異が最も端的に現われる CDS に入るデータのタイプと構造によって行うのが最適である。図-4 は現在考えられているクラス分けを示している。

以上述べた図-3 の論理モデルと図-4 のクラス分けは 1983 年 3 月までの検討結果であって、その後の検討は保留されており最終的に確定されたものではない。

1983 年 3 月以降は、論理モデルのような一般論に代わって基本クラスの検討が継続されている。この検討はほぼ収束しつつあるもののやはり最終的な確定には至っていない。以下に基本クラスのサービスとプロトコルを解説するが、上述のような状況にあることを注意されたい。なお、基本クラス以外のクラスの検討は着手されていない。

2. 基本クラスのサービス³⁾

2.1 機能の概要

基本クラスの仮想端末がユーザに提供する機能は論理モデルの CCA によって表現される。以下に、CCA の構成要素に従って説明する。

※ CDS (Conceptual Data Store) は基本クラスを特

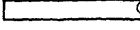
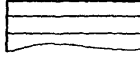
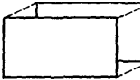
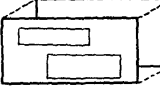
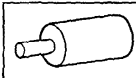
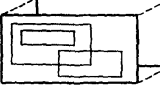
クラス	データ・タイプ	配置構造(X, Y, Z座標系)	実端末の例
基本クラス	キャラクタ・ボックス図形 	一次元  X: 有限/ 無限	テレタイプ ページモードディスプレイ
		二次元  X: 有限, Y: 有限/ 無限	
		三次元  X, Y: 有限, Z: 有限/ 無限	
フォームクラス	基本クラスと同じ	 2次元 / 3次元	データエントリ端末
イメージクラス	フォットグラフィック図形 	基本クラスの2次元または3次元と同じ	ファクシミリ
図形クラス	ジオメトリック図形 	 2次元 / 3次元	グラフィック端末
ミックスドモードクラス	上記の混在	図形クラスと同じ	高級ワークステーション

図-4 仮想端末のクラス案

色づけるとともに、サービスの中心的な役割を果たす。基本クラスのCDSは図-4内に図示されているようなキャラクタを配置するための構造体である。このような構造体を特定するパラメータには表-1に示すものがある。さらに、その構造体に対する操作には表-2に示すものがある。

両表に示すような特徴をもつCDSは、主に、編集済みテキストの表示処理に向けて設計されている。言い換えると、CDSにいったん入ったテキストを再編集するための機能は欠けている。たとえば、キャラクタ挿入や行削除を行う場合、挿入や削除によって配置位置が変わる無修正部分のキャラクタ列は、表-2を見ると、消去したのちに再設定という手順を踏まねばならない。これは明らかに処理時間の悪化を招く。

また、このCDSによって仮想端末プロトコルの実用化を図る場合、キャラクタ集合の機能キャラクタと

表-2の各操作の関係に注意する必要がある。テキスト操作では、CDSは属性制御操作で指定したキャラクタ集合のコードでテキストが作られていると理解す

表-1 基本クラス CDS の機能パラメータ

パラメータ	実 現 値
座 標 系	・次元数 (1次元, 2次元, 3次元) ・X, Y, またはZ座標の有限最大値または無限, など
キャラクタ・レパートリ	キャラクタ集合の数と集合の識別名
強 調	強調のレベル数とレベルの識別名
前 景 色	前景色の数と各色の識別名
背 景 色	背景色の数と各色の識別名
フ ォ ント	フォントの種類数と各種類の識別名
消 去	キャラクタ消去操作の可否

表-2 基本クラス CDS に対する操作

分 類	操 作 等	機 能 概 要
アドレッシング	暗黙的位置移動	配置要素にキャラクタを設定することによって、表示位置のX座標を1だけ増加させる。
	明示的位置移動	ユーザの指示で表示位置の移動を行う。newline, newpage, pointer absolute, pointer signed relative, および home がある。
内容更新	テキスト	表示位置の配置要素にキャラクタの設定を行う。
	属性制御	指定の属性を配置構造の指定範囲に設定する。属性には、キャラクタ・レバトリ、強調、前景色、背景色、およびフォントがある。
	消 去	指定範囲の配置構造のキャラクタを消去する。erase-line, erase-page, および erase-book がある。

るだけで、コードの意味を解釈しない。他方、表-2の各操作には CDS がその意味を理解できるコードが割り当てられている。したがって、同様の機能をもつ操作と機能キャラクタとを一緒に使用すると、CDS と実画面との間で図形キャラクタの双方の配置位置が異なるような矛盾が発生する恐れがある。

CSS (Control Signalling and Status Store) は図-3(b)に示すような端末の操作や運用機能のようなアプリケーションに大きく依存する機能に関連するため、CSS の標準化は深めることが不可能であるかも知しくはその必要がない。したがって、基本クラスは CSS を次の機能パラメータによって特定しているだけである。すなわち、データの種類 (キャラクタ、プリーアン等)、その大きさ、および後述するトリガ機構やトークン制御の有無である。これらのパラメータは実デバイスとの対応が可能な制御オブジェクトと称するエントリごとに設定される。

DSD (Data Structure Definition) は CDS や CSS の機能パラメータの実現値を保有する。DSD にパラメータの実現値が設定されたことは、仮想端末での通信環境が整ったことを意味する。すべての機能パラメータのうちの大半は、端末の種類に応じて実現値をあらかじめ決めておくことが可能である。このようなパラメータ実現値の集合はある種の端末の特徴を表現しているものであり、プロフィールと呼ばれる。基本クラスでは、いくつかのプロフィールが用意される予定である。

表-3 基本クラスのトークン

トークン	概 要	関連モデル要素
解放トークン	仮想端末コネクションの解放を開始する権利	CCA 全体
折衝トークン	仮想端末の使用条件を更新する権利	DSD
書き込みトークン	仮想端末のデータを更新する権利	CDS, CSS

AC (Access Control) はトークンと呼ぶ CCA へのアクセス権を所有しているユーザを記憶する。CCA が両ユーザの共用資源であることを思い起せば、CCA の内容を崩さないためにはトークンの所有はどちらか一方のユーザに限ることが必要となる。基本クラスのトークンの種類を表-3 に示す。

CSS の制御オブジェクトでトークン制御のないものは、書き込みが常に一方のユーザだけであることを意味する。CDS は、基本的にはトークン制御の対象であり、一時期には一方のユーザだけが CDS にデータの書き込みを行うことができる。双方のユーザが交互に書き込みを行うこと (すなわち会話) を可能とするため、トークンの移譲がユーザ間で行われる。

このようなトークン制御はプロトコルの観点からは送信権制御であり、トークンによる会話はいわゆる半二重通信に相当する。基本クラスの仮想端末は半二重通信をベースに概念が構築されているが、それと対照的な全二重通信も可能となるように概念の拡張が行われている (ただし、本稿では全二重通信に関する基本クラス仕様の説明は割愛する)。全二重通信とはいつでも会話型のデータ交換を行う場合は、なんらかの送信権制御がユーザの位置で必要となる。したがって、半二重通信と全二重通信の差は、主に、送信権制御を行う位置が仮想端末であるかユーザであるかである。仮想端末のインプリメンテーションでは、プログラム間インタフェース方式やメモリ設計などの点から両者の選択に対する配慮が必要である。

2.2 ユーザとのインタフェース

ユーザは 2.1 節で前述した機能をサービスプリミティブによって利用する。サービスプリミティブは機能を利用するための手段にはかならない。表-4 に基本クラスのサービスプリミティブの一覧を示す。

ユーザと CCA 間のサービスプリミティブの授受は SCP (Service Command Processor) によって制御される。図-3(c)に示されている SCP の諸制御機構が基本クラスではどのように規定されているかを以下に

表-4 基本クラスのサービスプリミティブ一覧

分 類	プリミティブ名	タイ プ	機 能 概 要
コネクションの確立	VT-CONNECT	確 認 型	コネクションの確立を行う。
コネクションの解放	VT-RELEASE	確 認 型	コネクションの通常の解放を行う。
	VT-DISCONNECT	非確認型	コネクションの緊急の解放を行う。
	VT-ABORT	指示だけ	サービス提供者によるコネクションの強制解放を行う。
使用条件の折衝	VT-START-NEG	確 認 型	折衝手順開始の同期制御を行う。
	VT-END-NEG	確 認 型	折衝手順終了の同期制御を行う。
	VT-NEG-INVITE	非確認型	使用条件提案の要請を行う。
	VT-NEG-OFFER	非確認型	使用条件の提案を行う。
	VT-NEG-ACCEPT	非確認型	VT-NEG-OFFER による提案の同意を行う。
	VT-NEG-REJECT	非確認型	VT-NEG-OFFER による提案の拒絶を行う。
データの転送	VT-DATA	非確認型	CDS および CSS を対象とするデータの転送を行う。
配 送 制 御	VT-DELIVER	非確認型	データ出力の指示を行う。出力確認の要請も行うことができる。
	VT-ACK-RECEIPT	非確認型	VT-DELIVER に対する確認の応答を行う。
トークン管理	VT-GIVE-TOKENS	非確認型	トークンの移譲を行う。
	VT-REQUEST-TOKENS	非確認型	トークンの移譲の要求を行う。

説明する。

アクセス制御機構はユーザのサービスプリミティブの発行可否を前述のトークンに基づいて判断する。基本クラスのトークンは表-3 が示しているように 3 種類のものがあるが、アクセス制御機構は 3 者間の区別を行わない。したがって、たとえば、一方のユーザがデータの書き込み (VT-DATA) を行っているときに、他方のユーザは折衝の (再) 開始 (VT-START-NEG) や通常の解放 (VT-RELEASE) を行うことはできない。

配送制御機構は、ユーザから CCA へ入力された VT-DATA を他方のユーザへどのように出力するかを規定する。配送制御機構には次の二レベルの利用法がある。

(1) 単純レベル ユーザが VT-DELIVER を発行すると、その時まで保留されていた VT-DATA は、もし存在したならば、出力される。なお、このような保留データの出力は VT-START-NEG, VT-GIVE-TOKENS, および VT-RELEASE に対しても行われる。

(2) 確認レベル 単純レベルに加えて、VT-DELIVER を受けたユーザは VT-ACK-RECEIPT を発行し、SCP はそれを他方のユーザへ渡すという手順が用意される。これにより、CDS または CSS にデータを書き込んだユーザは、そのデータが他方の

ユーザ (処理) に確実に反映されたことを確認できる。

トリガ機構は、ユーザの VT-DATA の発行を契機に、SCP がユーザに代って VT-GIVE-TOKENS を発行する制御機構である。

2.1 節で前述した CSS の制御オブジェクトがトリガ機構をもつ場合、上述のような制御が行われる。すなわち、このオブジェクトに対して VT-DATA が発行されると VT-GIVE-TOKENS の発行も SCP によって行われる。この機構は“データ入力キー”などを想像すると判り易い。このキーの押下はそれを抽象化した制御オブジェクトへの VT-DATA の発行を意味し、たとえば現カーソルポインタなどのデータとともに送信権が移譲される、というような譬えが可能である。

3. 基本クラスのプロトコル⁴⁾

基本クラスのプロトコル要素は対応するサービスプリミティブのパラメータに基づいて形成される。次の三とおりの方法がある。すなわち、プレゼンテーション層 (第 6 層) のサービスプリミティブのパラメータにただ変える方法、仮想端末が定める PDU(protocol data unit) を形成したのち第 6 層のプリミティブパラメータにする方法、および上記二つを合せて行う方法である。このように、基本クラスのプロトコル要素は第 6 層のプリミティブを利用して運ばれるわけである。

表-5 プレゼンテーション層サービスの利用概要案

VT サービス・プリミティブ	第6層利用法	VT-PDU	対応第6層プリミティブ	備 考
VT-CONNECT	直接および PDU 形成	—	P-CONNECT P-U-ABORT	左記第6層プリミティブ の複合によって手順を形成.
		CEQ/CER (connection establish)	P-TYPED-DATA	
VT-RELEASE	直 接	—	P-RELEASE	
VT-DISCONNECT	同 上	—	P-U-ABORT	
VT-ABORT	同 上	—	P-U-ABORT P-P-ABORT	P-U-ABORT は VT-PM が中断する場合.
VT-START-NEG	PDU 形 成	SNQ/SNR (start negotiation)	P-DATA	
VT-END-NEG	同 上	ENQ/ENR	同 上	
VT-NEG-INVITE	同 上	NIQ	同 上	
VT-NEG-OFFER	同 上	NOQ	同 上	
VT-NEG-ACCEPT	同 上	NAQ	同 上	
VT-NEG-REJECT	同 上	NJQ	同 上	
VT DATA	同 上	NDQ (controlled data)	P-DATA	CDS, および CSS の 制御オブジェクトに応じて 使い分け.
		UDQ (uncontrolled data)	P-TYPED-DATA P-EXPEDITED-DATA	
VT-DELIVER	直 接	—	P-SYNC-MINOR	
VT-ACK-RECEIPT	同 上	—	同 上	
VT-GIVE-TOKENS	同 上	—	P-GIVE-TOKEN	
VT-REQUEST-TOKENS	同 上	—	P-REQUEST-TOKEN	

VT-PDU の略号の終尾は, Q: request, R: response であることを示す.

が, その概要を表-5 に示す. 表-5 では, VT-PDU に要求 (request) と応答 (response) の2種類があることも同時に示している. VT-PDU の応答は仮想端末のサービスプリミティブが確認型である場合に限りて定められる. なお, 表-5 のプレゼンテーション層サービスの利用概要案は現在検討中である.

図-5 に仮想端末のサービスプリミティブと PDU, および第6層のプリミティブを使ったシーケンス例を示す. これは端末からホストコンピュータへの照会業務の場合である. 図-5 の配送制御を使った応答テキストでは, 受信側の VT-PM が P-DATA [NDQ] を P-SYNC-MINOR まで意図的に蓄積してもかまわないことも示している. このような蓄積は送信側の VT-PM が行ってもかまわない.

4. 今後の課題

第1章の最後で述べたように, 仮想端末の仕様はま

だ全般に不安定である. 約5年の歳月が経過してなおこの状態にあることにはさまざまな原因が考えられるが, 端末の進歩が急速なことで端末の機能はアプリケーションそのものであることが主な根本的原因であろう.

仮想端末プロジェクトは, このような状況をふまえて, ISO/OSI プロジェクトの組織変更を機会に標準化活動の見直しを検討している. 具体的には, 論理モデルとクラス分けに関する問題を明確にするとともに標準化活動の方法を提案することであり, 基本クラスの早期確定という課題に加え, 以下のような問題が挙げられている.

(1) 論理モデル関連

“端末管理”の扱いに関する問題である. 端末管理は, 端末オペレータのアプリケーションへのアクセス権, マルチウィンドウ制御, 端末へのデータ・プレロード等の機能の総称である. これは, 論理モデルで

