

森島秀実 竹内宏典
NTT情報通信処理研究所

1. はじめに

ワークステーション、パーソナルコンピュータなど、高機能化、低価格化の進む小型コンピュータを端末として用いることにより、これまでセンタで集中して行われてきた処理を、端末とセンタで分散して処理することが可能になってきた。このような垂直分散型システムを構築するためには、端末側に、センタとの通信を行いセンタが提供する機能を端末アプリケーションが利用可能にするための機能(MML機能)が必要となる。著者らが開発を進めている端末ミドルソフトは、このようなMML機能を実現し、端末上のアプリケーションに対してセンタや端末の種別に依存しないプログラムインタフェースを提供することにより、複数の異種センタに同時にアクセス可能とする。このようにして、端末からみた異種センタの機能の共通化を図る。

これまで著者らは、端末上のアプリケーションをセンタの環境に依存しないようにするために、センタにあるホストコンピュータ上のDB/DCパッケージ固有の処理手続き(ネイティブプロトコル)の仮想化を行った[1]。これによって、端末アプリケーションはプロトコルの種別を意識することなく、SEND/RECEIVEといった単純な手続きでセンタとの通信が行えるようになった。しかし、センタと端末を連携し、端末の使いやすさを生かしながらホスト資産を活用するためのMMLという観点では、通信だけではなく、より高度な機能(例えばデータベースへのアクセス機能)についてもセンタ環境に依存しないインタフェースが必要である。したがって、端末ミドルソフトにおいて、このような高度な機能(以下、応用機能と呼ぶ)の共通化を試みる。

本稿では2層からなるMML機能モデルの構成について述べ、そのモデルが実際のシステムに適用できることを明らかにする。最後に本モデルの有効性について考察する。

2. MML機能のモデル化

ここでは、これからのMML機能のモデルに求められる条件を明確にし、2階層からなるモデルの構成について述べ、そのモデルの動作についても言及する。

2.1 モデルが満たすべき条件

これからのMML機能モデルは以下の条件を満たすことが望まれる。

(a) センタの環境に依存しないこと

通信の種類、センタの機種、ホストのDB/DCパッケージが異なる場合でも、同一のモデルで、アプリケーションに対して共通インタフェースを提供することができなくてはならない。

(b) 種々の応用機能に適用できること

本モデルを用いて、様々なサービスを、各サービス毎に統一されたインタフェースで提供することができなくてはならない。さらに、サービスの追加に対しても容易に対応できなくてはならない。

2.2 MML機能モデル

応用機能の実現方式や端末から利用する際の手続きはセンタの環境により異なっているが、応用機能によって提供されるサービスは同じであると見なすことができる。例えば、オンライントランザクション処理を行うセンタに対して特定の電文を送信することによりセンタ側のアプリケーションが処理を開始するという点と、TSS型のセンタに対してプログラムの起動コマンドを送りセンタ側のアプリケーションが実行されるという点とは、センタアプリケーションを起動するという点で同じサービスと見なせる。しかし、同一サービスでもセンタによって利用手続きが異なるために、そのサービスを利用するアプリケーションはセンタ毎に異なっている。そこで、センタ毎に異なる手続きを処理する部分と、手続きの差を隠蔽する部分の2階層モデルとすることによって、アプリケーションに対してサービスを共通に提供することができる。以後、センタ毎に異なる手続きを処理する部分を応用通信層、手続きの差を隠蔽しサービスという形で共通にアプリケーションに見せる部分を応用サービス層と呼ぶ(図1-a)。ここで、応用通信層は、各センタ毎の通信手続きを記述したモジュールの集合から成り、応用サービス層は、アプリケーションに提供するサービス毎にモジュール化されている。

また、センタは端末から見るとサービスを端末に対して提供するためのサーバとして捉えることができる。センタをサーバとして捉えたとき、端末側は、サーバによって提供されるサービスを利用するクライアントとなる。上で述べた応用サービス層がクライアントとしての機能を果たす。つまり、アプリケーションがサービスを利用するとき、応用サービス層はクライアントとしてセンタ

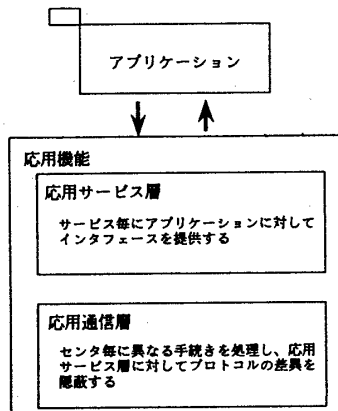


図1-a 応用機能の階層化

のサーバに対して要求を出す。要求は、応用通信の部分でセンタ毎の手続きに変換された後、センタでは要求されたサービスが実行される。その結果は、応用サービス層を通して、アプリケーションに返される。このようなクライアント/サーバモデルでは、アプリケーションは、センタの機種を意識することなく、クライアント（応用サービス層）を通してサーバ（センタ）を利用することができる（図1-b）。

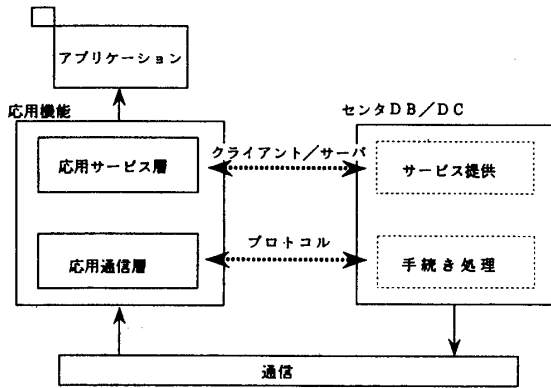


図1-b モデル構成

3. システムへの適応

本モデルが実際のシステムに適応することができることを確認するために、次のような考察を行った。

まず、センタが提供しているサービスを明らかにするために、NTTの計算機システムであるDIPS上のDB/DCパッケージに着目し、DB/DCパッケージの機能分析を行った[2]。その結果、サービスを以下の3つに大別することができた。

- (1) センタが端末向けに提供するサービス（例：リモートバッチ）
- (2) センタ間で利用するサービス（例：ファイル転送）
- (3) オフライン、あるいはオペレータ向けのサービス（例：コンソール制御）

これらを、端末ミドルソフトという観点から考察する。(1)は明らかに端末ミドルソフトでサポートすべきサービスである。(2)は端末の高機能化を考慮すると端末ミドルソフトでサポートする必要がある、すなわち、端末の高機能化により、センタと端末の関係は、従来の計算機-ダグム端末という関係から計算機-計算機という関係に変化してくるので、センタ間で用いられていたサービスも端末から利用できるようにしなくてはならない。それらに対し、(3)はセンタに閉じたサービスであるため、端末ミドルソフトのサポート対象外と考える。したがって、端末ミドルソフトで提供するサービスは表1に示すサービスとなる。

以下に、本モデルを用いてそれらのサービスを実現するための方法を示す。

表1のサービスのうち、端末向けタイプのサービスは、応用サービス層に通信の相手先の情報を持たせておき、アプリケーションが起動されセンタへのアクセスが要求されると、応用サービス層は相手先情報を参照し、それぞれのセンタに応じ

た電文を組み立て、センタに電文を送る。センタから返された各々の種類の電文を、応用サービス層がサービス毎に共通化してアプリケーションに返すという手法により実現される。

また、ファイルの転送、リモートAP起動等のサービスは、プロトコルによって実現される。本モデルではプロトコルは応用通信層でサポートし、応用サービス層では、そのプロトコルの操作を行うことによってファイル転送等のサービスを実現する。

さらに、モデルの拡張性について考えると、本モデルは応用機能を応用通信層、応用サービス層に分け、それぞれの各機能についてモジュール化を行って構成されている。したがって、新しいサービスの追加を行うとき、そのサービスを実現するモジュールを追加するだけで容易に行うことができることが分かる。

表1 サービスの分析

サービスの分類	サービスの概要	タイプ
DBアクセスサービス	データベースに対してアプリケーションからの要求に応じて参照、更新機能を実現し、その際の救済単位の設定、排他制御も行う	端末向け
ファイル転送サービス	他システム、他センタ間で各種ファイル情報をアプリケーションからの依頼を契機に送受する	計算機間
リモートバッチサービス	端末からの業務処理依頼を一旦センタで受付けた後、自動スケジュールにより処理し、端末から出力する	端末向け
リモートAP起動サービス	センタあるいは端末にあるアプリケーションを起動する	端末向け 計算機間
トランザクション通信サービス	コミットメント制御と同時実行制御を行なうことによりデータの一貫性を保証する	端末向け

4. おわりに

MML機能のアプリケーションから見たインタフェースを統合するモデルについて述べた。

その手法として、応用機能を応用通信層と応用サービス層の2層に分割し、それぞれについて機能毎のインタフェースの共通化を行った。こうして構築されたインタフェースを使うと、ユーザは共通化された応用機能を使ってアプリケーションを作成することができるため、アプリケーションを構築・保守することが容易になるという利点がある。

今回のモデルは、センタに应用機能を提供するためのサーバを仮定し、対応するクライアントを端末ミドルソフトソフトで実現することによって、アプリケーションに対してセンタ環境に依存しないサービスインタフェースを提供する方法を用いた。このような方法では、応用機能毎に端末ミドルソフトソフトをモジュール化することが可能であるため、機能拡張が容易になるという利点がある。

参考文献

[1] 菅沼毅：垂直分散システムにおけるホストネットワークプロトコルの仮想化：第41回情報処理学会全国大会, Sep 1990.
 [2] 佐藤亨, 中谷元, 伊藤路夫：APLICOTのAPインタフェース設計法：NTT R&D, Vol. 39, No. 11, p. 1565, 1990.