

解説

OSIの実現とその課題(IV)
ファイル転送, アクセス及び管理 (FTAM)



栗山幸造 野崎泰樹 笹島三郎

1. はじめに

今日、公共性の高い大規模システム、企業情報システムは多種多様なマルチベンダの情報処理関連機器(コンピュータ、端末、パソコンなど)、ネットワーク構成機器(モデム、PBX、マルチメディア多重化装置など)といったいわゆるC&C製品をCSMA/CD LAN, X.25 WANなどのOSI/相互運用ネットワークを介して接続し、n:n通信を基本とした高度な分散処理ネットワークシステムとして構築される傾向にある。

このような分散処理ネットワークシステムにおいては、おのおのの情報処理機器のもつ情報資源を有機的に結合し、相互利用することが非常に重要である。これを実現するためには、ネットワーク内のこれらの情報資源を機種に依存しない通信プロトコルにより容易に交換できることが必要である。このための手段の一

つに、仮想ファイルの概念を用いたデータ転送プロトコルとして現在国際的にもっとも注目を集めているFTAM(File Transfer, Access and Management)がある。図-1に、OSI/相互運用ネットワーク上でFTAMを用いてファイルデータを異機種システム間で共用するときの概念図を示す。

本稿では、FTAMのサービス/プロトコル仕様である基本標準の特徴、ならびにシステムの適用形態と実現すべき機能に応じ、基本標準で定められる各種の基本機能をサブセット化した実装規約の開発状況を概説するとともに異機種システム間およびC&C製品間でのファイル転送機能実現のためのFTAMの実装およびそこから派生した問題点について述べる。

2. FTAMの実装規約

2.1 FTAM基本標準概要

(1) FTAMのモデル

FTAMは、OSI応用層の一応用サービス要素であり、異機種システム間でのファイル転送(部分/全体転送)、ファイルデータ操作、ファイル属性情報の更新/参照を可能にしている。FTAMのモデルを図-2に示す。ここで、図中の“仮想ファイル”とは、実在するさまざまなファイルを送受信システム間で共通に理解できるように抽象化されたファイルのことである。

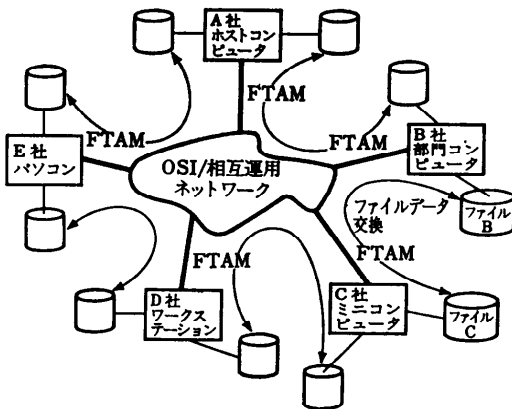


図-1 FTAMによるファイルデータの異機種システム間共有

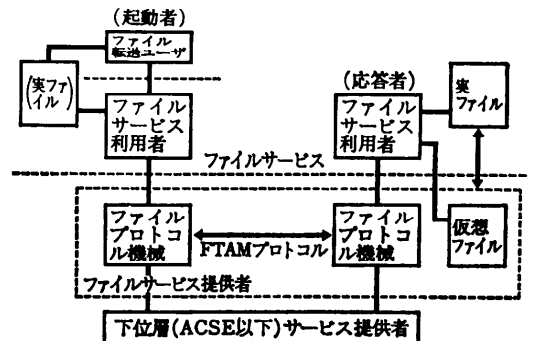
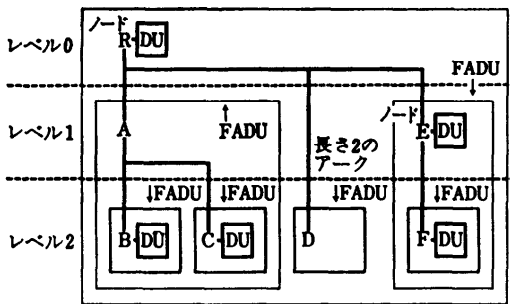


図-2 FTAMのモデル

† Implementations of Open Systems Interconnection (IV) File Transfer, Access and Management (FTAM) by Kozo KURIYAMA, Yasuki NOZAKI and Saburo KAGOSHIMA (Basic Software Development Division, NEC Corporation).

竹 日本電気(株)基本ソフトウェア開発本部



注) FADU: (File Access Data Unit)
 ファイルをアクセスするときの単位であり、
 ファイルの一部または全体を定義できる。
 アーク: 二つのノード間の連絡関係を示す。
 長さnのアーキとは二つのノード間のレベル
 差(距離)がnであることを示す。
 DU: (Data Unit)
 各ノードに対応付けられる
 データの集合。(DUの存在
 しないノードもある)

図-3 ファイルモデル¹⁾

る。FTAM は本仮想ファイルを対象として規定されるサービス/プロトコル群より構成される。

(2) FTAM で扱えるファイル

FTAM は、さまざまなファイルを想定して、木構造をなすファイルモデルを定義している(図-3 参照)。ファイルへのアクセスは、木構造の一部または木構造全体に対して行うことができる。

(3) FTAM サービス

FTAM サービスプリミティブ一覧を表-1 に示す。

2.2 FTAM 実装規約の開発

(1) 実装規約の考え方

FTAM を各システムに実装する際の規約(実装規約と呼ぶ)の開発は、以下のことを考慮して行われている。

FTAM 基本標準には、適用ネットワーク/アプリケーション環境の多様性を考慮して数多くの機能がオプションとして規定されている。しかし、機能/性能の異なるさまざまなシステムとの相互接続を考慮すると、FTAM 基本標準に規定されて

いるすべての機能を実装することは、プログラムの規模/実行性能面などから考えて、現実的でない。これを避けるために、システムの能力や利用者からの要求条件を考慮し、FTAM 基本標準で規定されている機能の一部を実装することになる。

この場合、個々のシステムで勝手に機能選択をしたのでは、相互接続性が失われるため、機能切り出しのためのガイドラインが必要となる。具体的には、パラメタ長/値の範囲、オプションパラメタの選択(使用する/しないか)を規定することが必要となる。

また、FTAM 基本標準の機能の切り出しだけでなく、FTAM が使用するプレゼンテーション層/セッション層の機能の切り出しも行われる。

(2) 実装規約での規定事項³⁾

実装規約として以下のような項目の規定が行われて

表-1 FTAM サービスプリミティブ一覧³⁾

サービス項目		プリミティブ	
FTAM レジーム制御	FTAM レジーム確立 FTAM レジーム終了 FTAM レジーム中断 "	F-INITIALIZE F-TERMINATE F-U-ABORT F-P-ABORT	確認型 確認型 非確認型 非確認型 (提供者起動)
ファイルストア管理 注)1	—	—	—
ファイル選択 レジーム制御	ファイル選択 ファイル解放 ファイル生成 ファイル削除	F-SELECT F-DESELECT F-CREATE F-DELETE	確認型 確認型 確認型 確認型
ファイル管理	属性読出し 属性変更	F-READ-ATTRIB F-CHANGE-ATTRIB	確認型 確認型
ファイルオープン レジーム制御	ファイルオープン ファイルクローズ	F-OPEN F-CLOSE	確認型 確認型
グループ化制御	グループ化開始 グループ化終了	F-BEGIN-GROUP F-END-GROUP	確認型 確認型
ファイル内容 アクセス	FADU 位置付け FADU 削除	F-LOCATE F-ERASE	確認型 確認型
バルクデータ転送	読出し 書込み データ単位転送 データ転送終了 転送終了 転送キャンセル	F-READ F-WRITE F-DATA F-DATA-END F-TRANSFER-END F-CANCEL	非確認型 非確認型 非確認型 非確認型 確認型 確認型
回復 注)2	レジーム回復	F-RECOVER	確認型
チェックポイント 再開 注)2	チェックポイント データ転送再開	F-CHECK F-RESTART	確認型 確認型

注) 1: 将来追加が予定されている。
 注) 2: 内部ファイルサービス選択の場合のみ。

いる。

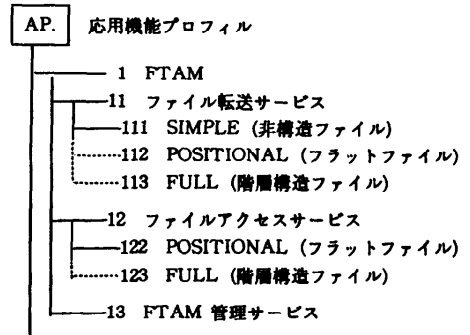
- (a) サポートレベル (実装必須/オプション/未実装) の定義
 - (b) 仮想ファイルの各属性のサポートレベル
 - (c) サービスプリミティブおよびそのパラメタのサポートレベルの定義
 - (d) 仮想ファイル属性/サービスプリミティブパラメタ/PDU パラメタの取りうる値と既定値などの詳細規定
 - (e) ACSE/プレゼンテーション/セッションへのマッピング方法
 - (f) 診断コードとサービスプリミティブの関係
 - (g) アクティビティ属性のパラメタ設定方法
 - (h) ドキュメントタイプの規定
 - (i) FTAM-PDU の符号化方法
- (3) 実装規約の国際協調

FTAM 実装規約化の作業は、INTAP (情報処理相互運用技術協会: 日本), SPAG (Standards Promotion and Application Group: 欧州), NIST* (National Institute of Standards and Technology: 米国) などの OSI 推進機関を中心として行われており、それぞれ実装規約書を作成していた。その際、互いに相互接続可能なように、上記の規定内容をお互いに合わせるような努力をしてきた。しかし、最近になって ISO/IEC JTC 1 で実装規約を作成することになったため、今後は、JTC 1 の場で実装規約書 (正式には、国際標準プロファイル (ISP) と呼ばれる。) が作成されることになった。なお、今までに各機関で作成されてきた実装規約書の内容は、ISP の中にも矛盾のない形で含められる予定である。

2.3 実装対象 FTAM プロファイル

(1) 実装規約の選択

FTAM の実装規約は 2.2 (3) 実装規約の国際協調で述べたごとく、国際協調のもとに開発されてきており、現在 JTC 1 の場で ISP として開発中である。ISP と整合化され INTAP において開発対象となる機能サブセット群 (プロファイルと呼ぶ) は図-4 のような構成となっている。ここで AP. 111 (非構造ファイルの一括転送のためのプロファイル) については ACSE, プレゼンテーション層の IS 化対応作業を残し現在開発を完了している。また他のプロファイルについては現在開発中または開発検討中にある。



注) 数字はプロファイル番号を示す。

図-4 FTAM 関連のプロファイル^{*)}

AP. 111 は国内はもとより国際的にももっとも高度にハーモナイズされており、昨年 11 月の INE'88 において、本規約に基づいた国内/海外相互接続デモが、ISO 8802-4 トークンバス LAN, X. 25 WAN の各ネットワーク上で実施された。また市場において特に要求の強いファイルの転送を実現する規約であることなどの状況、理由により本稿では実装対象とする規約については INTAP/AP. 111 を採用した。

(2) AP. 111 プロファイルの概要⁶⁾

FTAM の実装規約の規定事項に従い、AP. 111 プロファイルの実装規定内容を示すとともに、FTAM のモデルへの実装マッピングを示す。

(a) サポートレベル

FTAM の実装規約は仮想ファイル、サービス、プロトコルのおのおのに関し、基本標準で記述された各項目について、サポートの選択を記述している。サポートのレベルは以下に示す "m", "s", "os", "—" の四種類に大別される。各項目はこのいずれかのレベルでサポートされるものとしてレベルが選択される。

m (mandatory: 必須) : 本項目は、基本標準で必須とされる。したがって本実装規約に適合するすべての FTAM 製品に実装されなければならない。

s (supported: サポート) : 本項目は、本実装規約においてサポートすると規定したため、各 FTAM 製品は、この規約に適合するためには実装しなければならない。この項目をサポートしない実装とも折衝により項目をはずすことにより相互動作しなければならない。

os (optionally supported: オプションサポート) : 本項目をサポートするか否かは各 FTAM 製品に任せられる。

— (not supported: サポートしない) : 本項目は本

* 以前の NBS (National Bureau of Standards)。昨年、名称変更された。

実装規約の範囲外である。

(b) 仮想ファイル、サービス、プロトコルのサポートレベルと実装マッピング

仮想ファイル、サービス、プロトコルの基本標準における各機能項目に関し、AP. 111 でのサポートレベル、FTAM のモデルにおける実装レベルを示す。

ここで表-2 に示す FTAM, FTF とはそれぞれ図-2 の FTAM のモデルにおけるファイルプロトコル機械、ファイルサービス利用者に対応する実装プロダクトである。実装上の考え方としては基本的には FTAM は AP. 111 以外のファイルアクセス (AP. 122) / ファイル管理 (AP. 13) などの実装規約に準拠

するプロダクトに対して対応可能とするため、全機能項目を実装対象とする。また FTF に関しては AP. 111 に準拠したファイル転送利用者機能範囲を実装対象とする。(表-2 参照)

(c) ACSE, プレゼンテーション, セッションの使用法とサービスプリミティブのマッピング

- ACSE の使用法

A-ASSOCIATE, A-RELEASE, A-ABORT, A-P-ABORT を使用する。

- プレゼンテーションの使用法

使用される機能単位をサポートレベルを示す。

カーネル

m

コンテキスト管理 —

コンテキスト復旧 —

- セッションの使用法

使用される機能単位をサポートレベルを示す。

カーネル m

全二重 s

小同期 OS

再同期 OS

その他の機能単位 —

- サービスプリミティブのマッピング

FTAM, ACSE, プレゼンテーション, セッション間のサービスプリミティブのマッピングに関し、表-3 に示す。

表-2 仮想ファイル、サービス、プロトコルのサポートレベルと実装マッピング例 (抜粋)

			FTAM 基本標準の機能項目	AP. 111 のサポートレベル	FTAM の実装レベル	FTF の実装レベル		
仮想ファイル動作	ファイル全体への動作	ファイル生成		os	○	○		
		ファイル選択		m	○	○		
		属性変更		—	○	—		
		属性読出し		os	○	○		
		ファイルオープン		m	○	○		
		ファイルクローズ		m	○	○		
		ファイル削除		os	○	○		
		ファイル解放		m	○	○		
		~~~~~						
		サービス	サービスクラス	ファイル転送クラス		s	○	○
ファイルアクセスクラス				—	○	—		
ファイル管理クラス				—	○	—		
無制限クラス				—	○	—		
転送・管理クラス				—	○	—		
~~~~~								
プロトコル	ファイルアクセス	F-LOCATE		—	○	—		
		F-ERASE		—	○	—		
	転送データ	F-READ		s	○	○		
		F-WRITE		s	○	○		
		F-DATA		m	○	○		
		F-DATA-END		m	○	○		
		F-TRANSFER-END		m	○	○		
		F-CANCEL		m	○	○		
	チャク再エリ開	F-GHECK		os	○	○		
		F-RESTART		os	○	○		

サポートレベル: m (必須) s (サポート) os (オプションサポート) — (サポートしない)
 実装レベル: ○ (実装する) — (実装しない)

3. FTAM の実現

3.1 実装上の方針

これまでの各社の独自 NA (Network Architecture) は、主に情報処理関連機器相互間の接続を対象領域としていた。今日、情報処理関連機器、ネットワーク構成機器を含めた異機種システムの自由な接続を目的として開発された OSI の出現により、オープン性、接続領域が拡大されることになり、各社の NA は OSI を拡張 NA として取り込む傾向にある。このような NA の統合化傾向に対して、それを実装する製品が機種ごと異なっていたので

表-3 サービスプリミティブのマッピング

FTAM	ACSE	プレゼンテーション	セッション
F-INITIALIZE	A-ASSOCIATE	P-CONNECT	S-CONNECT
F-TERMINATE	A-RELEASE	P-RELEASE	S-RELEASE
F-U-ABORT	A-ABORT	P-U-ABORT	S-U-ABORT
F-P-ABORT	A-P-ABORT	P-P-ABORT	S-P-ABORT
F-SELECT		P-DATA	S-DATA
F-DESELECT		P-DATA	S-DATA
F-CREATE		P-DATA	S-DATA
F-DELETE		P-DATA	S-DATA
F-READ-ATTRIBUTE		P-DATA	S-DATA
F-OPEN		P-DATA	S-DATA
F-CLOSE		P-DATA	S-DATA
F-BEGIN-GROUP		P-DATA	S-DATA
F-END-GROUP		P-DATA	S-DATA
F-RECOVER*		P-DATA	S-DATA
F-READ		P-DATA	S-DATA
F-WRITE		P-DATA	S-DATA
F-DATA		P-DATA	S-DATA
F-DATA-END		P-DATA	S-DATA
F-TRANSFER-END		P-DATA	S-DATA
F-CHECK*		P-SYNC-MINOR	S-SYNC-MINOR
F-CANCEL		P-RESYNCHRONIZE (abandon) or P-DATA	S-RESYNCHRONIZE (abandon) or P-DATA
F-RESTART*		P-RESYNCHRONIZE (restart)	S-RESYNCHRONIZE (restart)
		P-TOKEN-GIVE	S-TOKEN-GIVE

* ファイルサービス利用者からは直接利用できない。
誤り回復プロトコル機械により提供されるサービスプリミティブ。

は、OSI の普及の足かせとなる。そこであらゆる機種に効率的かつタイムリに同様な機能を提供するためには、分散処理アプリケーション構築基盤を提供する API (Application Program Interface) ならびにプロトコル実装の共通化が非常に重要となる。

(1) 機種独立な FTAM/API (アプリケーションインタフェース) の開発

(a) 狙い

FTAM/API は、以下のことを狙いとして設計を行う。

- アプリケーションの機種間共通性の向上
機種共通なインタフェースを提供することにより、アプリケーションの移植および移行を容易にする。
- アプリケーション開発の生産性の向上
高級言語 (C, COBOL) での API の提供、各 API 間で類似のアクセス方式の採用、複雑な通信プロトコルを隠した簡易な API の提供によりアプリケーション開発の生産性を向上させる。

(b) FTAM/API

FTAM/API を以下の C 言語関数として定義する。

• FTAM 関数

[定義]

```
#include <osi.h> /*OSI-AP 共通ヘッダファイル*/
#include <ftam.h> /*FTAM ヘッダファイル*/
int Ftam (selector, arg);
short selector;
struct _ftam_s *arg;
引数 1 : 機能セレクト
```

セレクト名	機能
FmOPN	開始機能
FmCLS	終了機能
FmPUT	送信機能
FmGET	受信機能
FmCLSec	終了機能の完了情報取得機能
FmPUTcc	送信機能の完了情報取得機能
FmGETcc	受信機能の完了情報取得機能

引数 2 : FTAM 構造体へのポインタ

FTAM 提供者と FTAM 利用者の情報交換に用い

```

/*****
/*      サービスプリミティブ 非同期受信処理 ルーチン (例一抜粋)      */
/*****
#include <osi.h>
#include <ftam.h>
/*****
void function ( ftam_sp )
struct _ftam_s  *ftam_sp;
{
    struct _mb_param  mb_param;
    short             rc;

    :
    :
    ftam_sp -> messagep -> msgid=MsgGET;
    :
    :
    switch ( rc=Ftam( FmGET, ftam_sp )) /* 受信処理要求 */
    {
        case Rc_NORMAL : break ;
        case Rc_WARNING : ..... ; break ;
        case Rc_FATAL   : ..... ; break ;
        default        : ..... ; break ;
    }
    switch ( rc=MBox( MbWAIT, &mb_param )) /* メッセージ待合せ要求 */
    {
        case N_NMLFIN:
            switch ( mb_param. msgid )
            {
                case MsgPUT : /* メッセージ=PUT 完了情報 の場合 */
                    switch ( rc=Ftam( FmPUTcc, ftam_sp ))
                    { ..... ; /* PUT 完了情報取得処理 */ }
                    break ;
                case MsgGET : /* メッセージ=GET 完了情報 の場合 */
                    switch ( rc=Ftam( FmGETcc, ftam_sp ))
                    {
                        case N_NMLFIN : switch ( ftam_sp -> fsp )
                            { case F_INIID :
                                ..... /* F-INITIALIZE ind 受信処理 */;
                                break ;
                                case F_SELID :
                                ..... /* F-SELECT ind 受信処理 */;
                                break ;
                                default : ..... ; break ;
                            }
                        break ;
                        case F_PARAER : ..... ; break ;
                        case F_SYSER  : ..... ; break ;
                    }
                break ;
            }
        case F_PARAER: ..... ; break ;
        case F_SYSER : ..... ; break ;
        default     : ..... ; break ;
    }
}
/*****

```

図-5 FTAM/API を利用した AP 例

られるデータ構造体

関数値：リターンコード

- 非同期アクセス関数

[定義]

```
#include <osi.h>
int MBox (selector, arg);
short selector;
struct _mb_param *arg;
引数 1：機能セレクト
```

セレクト名	機能
MbALLOC	メッセージボックス生成機能
MbREMOVE	メッセージボックス削除機能
MbWAIT	複数事象待合せ/メッセージ取得機能
MbNOTIFY	メッセージ登録機能

注) メッセージボックス：非同期メッセージ格納域

引数 2：メッセージボックスアクセス構造体へのポインタ

メッセージボックスをアクセスするときの入出力情報

関数値：リターンコード

本 FTAM/API を利用した AP の例を図-5 に示す。

なお、MAP においても FTAM の C 言語によるランゲージバインディングが試みられている⁹⁾。

(2) 機種独立な FTAM プロトコルの実装

FTAM プロトコルの機種を越えた実装の共通化は API の共通化とともに製品開発における重要な要素である。そこで、FTAM を C&C 製品の各機種共通に実装するための問題点とその解決方法について述べる。

(a) 問題点

- FTAM プロトコルの実装/動作環境 (ジョブ空間またはシステム空間、分配機能または非分配機能など) が OS ごとに異なる。

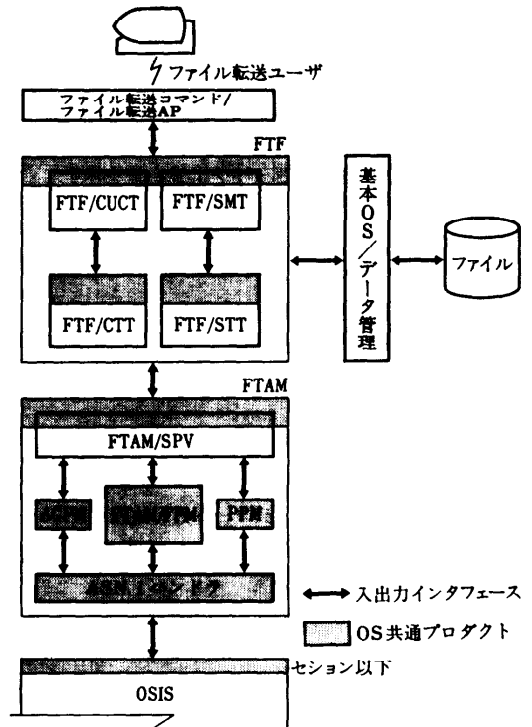
- FTAM プロトコル実装で使われる OS 基本機能 (メモリ管理、バッファ管理など) に関してもその利用上のインタフェース/機能が異なる。

- 移植性の高い C 言語の仕様に関してもアーキテクチャ依存の差異が残る。

(b) 解決方法

- OS 共通プロダクトと OS 固有プロダクトに分けて開発を実施する。

OS 共通プロダクト：機種間で完全可搬性のあるプロダクトで本部分が大きくなるような設計上の工夫を



- FTAM (FTAM 通信ソフトウェア)
- FTAM/FPM (FTAM のプロトコル処理モジュール群)
- ACPM (ACSE のプロトコル処理モジュール群)
- PPM (プレゼンテーションのプロトコル処理モジュール群)
- ASN. 1 ハンドラ (ASN. 1 抽象構文/転送構文変換モジュール群)
- FTAM/SPV (FTAM ディスパッチャモジュール群)
- OSIS (セッションのプロトコル処理モジュール群)
- FTF (ファイル転送フェシリティ)
- FTF/CUCT (FTF のクライアント利用者制御モジュール群)
- FTF/CTT (FTF のクライアント転送機能モジュール群)
- FTF/SMT (FTF のサーバ利用者制御モジュール群)
- FTF/STT (FTF のサーバ転送機能モジュール群)

図-6 FTAM ファイル転送システム

行う。

OS 固有プロダクト：機種ごとに異なり可搬性のないプロダクトで本部分が小さくなるような設計上の工夫を行う。

- 移植性の高いシステム記述言語である C 言語を開発言語とし、機種間共通利用範囲のルール化を実施する。

以上で述べた方法に従い FTAM プロトコルを実装するソフトウェアプロダクトに関し、OS 共通プロダクト、OS 固有プロダクトへの分割を実施する。

図-6 に FTAM プロトコルを実装するファイル転送システムのソフトウェア構成を示す。

(c) OS 共通プロダクト

入出力インタフェース機能, プロトコル処理機能, PDU のエンコード/デコード機能などが共通化対象である。以下, 対象プロダクトと共通機能を示す。

- FTAM
入出力インタフェース機能など
- FTAM/FPM
入出力インタフェース機能
FTAM プロトコル処理機能
FPDU のエンコード/デコード処理機能など
- ACPM
入出力インタフェース機能
ACSE プロトコル処理機能
APDU のエンコード/デコード処理機能など
- PPM
入出力インタフェース機能
プレゼンテーションプロトコル処理機能
FPDU のエンコード/デコード処理機能など
- ASN. 1ハンドラ⁹⁾
入出力インタフェース機能
ASN. 1エンコード/デコード処理機能など

なお, FTAM ファイル転送システムを構成する FTAM 以外のプロダクトに関しても図-6 に示すごとく共通化を実施した。

(d) OS 固有プロダクト

ディスパッチャ機能などの共通化対象外の機能であり, 以下のプロダクトよりなる。

- FTAM/SPV
FPM/ACPM/PPM 各プロトコル処理へのディスパッチャ機能
FTAM アクセスインタフェースユーザ制御機能
セッションアクセスインタフェース制御機能など

(e) Cコーディング規則

機種間共通性を保証するためにCコーディング使用規則を設定した。

3.2 実装方式/技術

FTAM を利用したファイル転送システムの実装ソフトウェア構成の概要を図-6 に示した。

ここでは FTAM を実現する上での中核機能に位置付けられる OS 共通プロダクトである FTAM/FPM に関し, その実装ソフトウェア構成の詳細を示すとともに各 OS の実装/動作環境に依存しない実

装方式/技術について説明する。

(1) 実装ソフトウェア構成

(a) コンポーネントの位置付け

FTAM/FPM は FTAM を構成する機能の一つであり, 各 OS 共通部品としての機能モジュールである。図-7 に FTAM/FPM の位置付けを示す。

(2) OS 間共通実装方式/技術

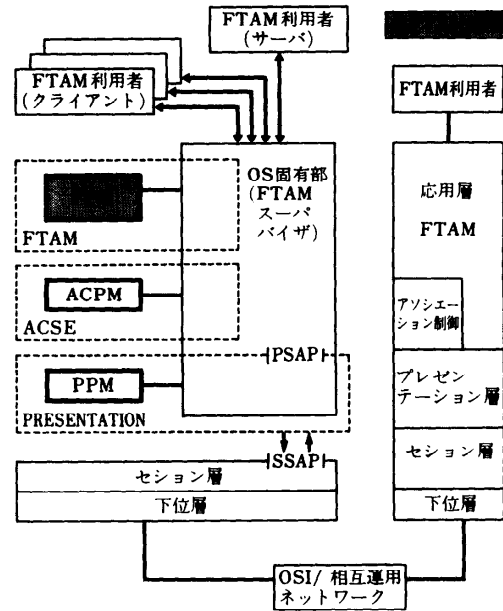
FTAM/FPM は異 OS 間での共通利用性を保証するために仮想化されたインタフェース方式を採用する。本インタフェースはC関数として実現され, 異なる OS 環境下での利用を可能とする。本関数により提供される機能の要旨は以下の3点である。

- FTAM プロトコル処理 (状態遷移処理, FPDU の生成/分解処理) の実行
- 下位層 (ACSE またはプレゼンテーション) サービスプリミティブとその引数の参照 (受信) /生成 (送信) 処理の実行
- 次に FTAM/FPM 利用者が実行すべき動作スケルトン (=動作シナリオ) を関数値として返却する。

以下, インタフェースの概要を説明する。

(a) 利用者インタフェース (=関数)・機能

① fpmput



□ : OS 共通プロダクト

図-7 FTAM/FPM の位置付け

FTAM サービスプリミティブとその引数を入力とし、FPM 利用者が次に行うべき動作指示を値として返す。その際、以下の処理が実行される。

- FTAM サービスプリミティブの正当性検査
 - 引数の正当性検査
 - FPDU (ftam-pci, ftam-fadu, fcde) の生成 (エンコード)
 - 下位層 (ACPM, PPM) サービスプリミティブとその引数構造体生成
 - 状態遷移
- ② fpmget

下位層サービスプリミティブとその引数を入力とし、FPM 利用者が次に行うべき動作指示を値として返す。その際、以下の処理が実行される。

- FPDU の分解 (デコード)
- 下位層発生イベントの正当性検査
- 引数の正当性検査
- 状態遷移

表-4 に fpmput, fpmget の関数値 (抜粋) を示す。

③ FPM 構造体

FPM 利用者と FPM の間で情報を受け渡すために用いる構造体である。図-8 に本構造体の概要を示す。

(b) 利用シーケンス

本インタフェース方式を使用した利用者の動作イメージに関し、その概要を図-9 に、またファイル転送 (Write) を例にとり、その詳細を図-10 に示す。

3.3 FTAM の移植

上記 3.1~3.2 で述べた方針、方式/技術に基づき第一次適用機種上で開発した FTAM を、第二次適用機種上に移植/展開を試みた。以下に本経験をもとに移植に関する考慮点などについて述べる。

(1) PDU が各プロトコルマシン部を通過する際、PDU のバッファ間移送をともなう実装方式では、バッファ間移送による性能低下が問題となる。

この回避の方法としては、FTAM の利用者データと各層のプロトコルヘッダ部をそれぞれ一回の作成移送ですむように工夫した。

(2) 多重化処理を FTAM の一つのプロトコルマシン部では処理せずに、各ユーザごととその延長上でプロトコルマシン部を動作させることにより、FTAM 自身による多重化制御が不要となる。また、リソース

表-4 FPM 関数値一覧 (抜粋)

関数値定義マクロ名	値	FPM 利用者動作概要
① FmRTFSU	0×0001	内部ファイルサービス利用者に戻る。
② FmP_ACPM	0×0002	ACSE にすべての FPDU がなくなるまで送信要求を実行する。
③ FmP_PPM	0×0003	PPM にすべての FPDU がなくなるまで送信要求を実行する。
④ FmP_PPMdiscard	0×0004	PPM にすべての FPDU がなくなるまで送信要求を実行する。ただし、最後の要求は破棄された。
⑤ FmG_PPM	0×0005	PPM に受信要求を実行する。
⑥ FmPG_PPM	0×0006	PPM に送信要求後、受信要求を実行する。

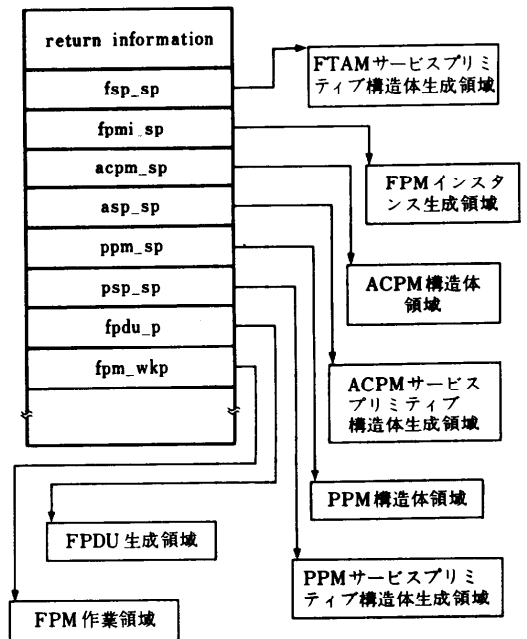


図-8 FPM 構造体形式

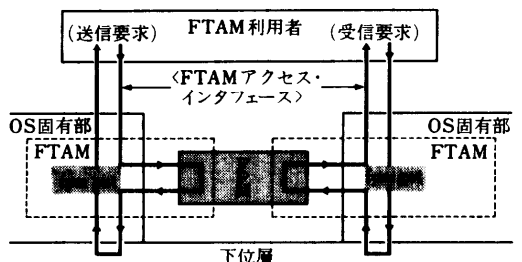


図-9 FTAM/FPM 利用概要

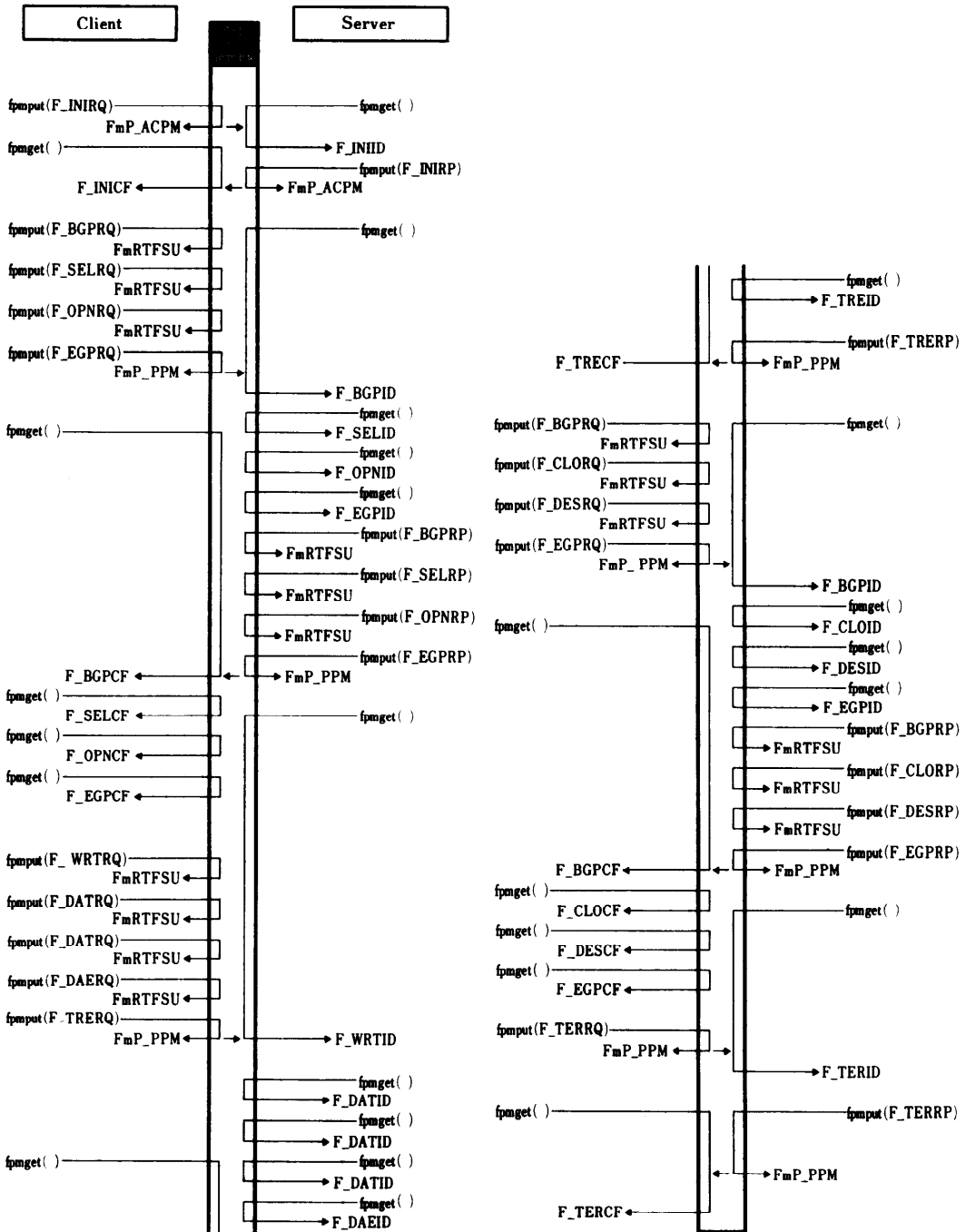


図-10 ファイル転送 (Write) 利用シーケンス

が各ユーザ空間に従属されるため、ユーザ数の制限が論理的には、なくなるといった利点がある。

(3) FTAM の各機種間の移植の検討・推進と合

わせて重要な要素として、試験システムの移植あるいは流用があげられる。本検討では、各機種間でテストデータ・シナリオに基づき、FTAM のプロトコル/

サービスの評価が均一に実現できる FPT (FTAM Protocol Tester) を開発した。テストデータ・シナリオは全機種に適用され、評価手続き部のみ移植コーディングを行うだけでよく、きわめて高い生産性と FTAM の機能と品質の均質性が得られている。

4. FTAM の課題と関連プロトコルとの比較

4.1 FTAM 基本標準の課題/改良点

FTAM のように大容量のファイルデータをネットワークを介して転送するプロトコルの場合、転送性能、障害からの復旧/再開能力、さまざまなファイルデータ形式の交換可能性がすぐれているか否かが問題となる。

まず転送能力については、バルクデータ転送機能、PDU の連結機能などデータ転送を効率よく実行できるプロトコルメカニズムを保有しているといえるが、一方データの圧縮転送という点からみるとプレゼンテーション層の機能も含めて効果的な一般メカニズムがまだ開発されていない。また、ASN.1 符号化法に基づく PDU の表現方法も自由度が非常に高く、それが逆に受信側での PDU のデコード時の負担となり性能に悪影響を及ぼす原因にもなっている。実装規約などで表現方法の制約を設ける必要がある。

次に障害からの復旧/再開能力であるが、標準機能として任意のチェックポイントから復旧/再開ができるリカバリ、リスタートプロトコルをもっており、高い復旧/再開能力をもつプロトコルといえよう。

最後にファイルデータ形式の多様な表現性であるが、FTAM ではファイルの構造、データ形式をドキュメントタイプにより定義するのが一般的である。しかし現在定義されているドキュメントタイプは単純なデータ形式を表現するにとどまっております、複雑なデータ形式(たとえばマルチメディアデータ)を表現することには難がある。今後の検討テーマである。

4.2 他の類似プロトコルとの比較

異機種間でのファイル転送/アクセスプロトコルとして FTAM の他に、米国防務省で開発された FTP (File Transfer Protocol)、AT & T で開発された RFS (Remote File Sharing)*、Sun Microsystems 社の NFS (Network File System)* がある。FTP と FTAM の比較は文献¹⁰⁾に述べられているので、

ここでは割愛する。RFS と NFS は、分散ファイルシステムが構築できるように設計されており、現在大いに注目されている。RFS と NFS の大きな違いは、RFS が、サーバ側で状態を管理するのに対し、NFS はサーバ側で状態を管理しないという点、また RFS は UNIX 間のファイル共有を目的としており、UNIX との密着性が高い点などである。その他の機能については、両者ほぼ同等の機能を有す。

以下、RFS、NFS と FTAM との比較を行う。

(1) クライアント/サーバモデル

RFS、NFS のプロトコルは、クライアント/サーバモデルをベースに設計されており、この点では FTAM と対等の関係にある。しかし、RFS、NFS にはファイル転送機能がなく、ディレクトリのリモートマウントに基づくファイルアクセス機能を提供するものであり、FTAM が規定している回復、再開、チェックポイントなどのサービス機能はもたない。

(2) ファイルアクセス構造

FTAM においては、ファイルアクセス構造として階層型、非構造、フラットの三種類があり、扱えるファイルとして順編成、直編成、索引順編成などを想定しているが、RFS、NFS の場合は単なるバイトの集合としてファイルを扱っており、非構造ファイルを対象としている。

(3) ファイルシステムサービス

RFS、NFS は OS ファイルシステムによりローカル/リモートファイルを、利用者が区別することなくアクセスするサービスを提供している。したがってすべての利用者は分散不可視でファイルの使用が可能である。一方、FTAM にもファイルアクセス管理サービスがあるが、RFS、NFS とは提供機能レベルが異なり、分散不可視機能がない。したがって、利用者に分散不可視なサービスを提供するには、OS としてファイルディレクトリ管理機能をサポートする必要がある。

また、RFS、NFS は階層ディレクトリ機能を有している。FTAM では、同等な機能 (FTAM ではファイルストア管理機能と呼ぶ) を現在検討中である。

5. おわりに

機器の多様化、ネットワークの広域化、複数化、高度化がますます促進される今日の高度情報社会においては、情報処理関連機器、ネットワーク構成機器などのいわゆる C & C 製品の相互接続を可能とし、その

* RFS は AT&T の登録商標であり、NFS は Sun Microsystems 社の登録商標である。

上で高度な情報処理機能を提供する国際標準プロトコルである OSI の普及は不可欠である。

その中でも、FTAM はエンドシステム間でもっとも基本的な情報交換手段を提供するものであり、しかも、基本標準と実装規約の開発がもっとも進んでいるために、この FTAM がどれだけ広く普及するかが OSI の将来を左右するといっても過言ではない。

本稿ではファイルデータの一括転送に関する実装規約の開発と製品への共通実装方式の一例について紹介した。

FTAM プロトコルの適応領域を拡大するためにはさらにリモートファイルアクセスシステム、リモートファイル管理システムなどの拡張機能向け実装規約の開発ならびにその規約の製品への実装が必要となる。これらの拡張領域はファイル転送と異なり、実システム環境 (RSE) のファイル管理システムと OSI 環境 (OSIE) の FTAM/仮想ファイルストアとの密接な対応付け、変換を必要とするため、実装規約設定の範囲、実装方式はさらに複雑かつ高度となる。これまで同様、規約の設定に関しては国内/国際的調整を密に行い、将来にわたって利用度の高い仕様の開発を行っていく必要がある。

また、FTAM は現在 R&D、デモンストレーションその他の場をとおして実際面への適用性の検討フェーズにある。次年度にもなると実製品として市場に出され各種のネットワークシステムに利用され、評価をうけることとなる。OSI 自体が独自性を排除し、オープン性を追求することから、公共性の高い大規模システム、マルチベンダの製品で構成される大規模企業情報システム、LAN 環境などが主要適用領域として検討されており、プロトコルも純粋に OSI のみで構成する状況にある。本傾向は OSI の役割からみて将来的にも変わらないと考えられる。しかしながら、メーカ独自のネットワークアーキテクチャや業界標準プロトコルを基盤として構築されたネットワークから OSI ネットワークへ移行する際、過渡的段階として FTAM に関しても、対応する独自プロトコルとの相

互変換のためのゲートウェイ機能が必要となる。この場合のゲートウェイは各プロトコルデータユニット (PDU) の 1 対 1 変換により実現されるというよりは、ファイルデータのエンドシステム間での交換を実現することに主眼をおいたユーザプロセスレベルでのファイルデータのトランスペアレントな中継処理により実現されることとなる。応用面でのさらに一層の検討が必要である。

参考文献

- 1) ISO 8571-1, Information Processing Systems—Open Systems Interconnection—File Transfer, Access and Management—Part 1: General introduction.
- 2) ISO 8571-2, Information Processing Systems—Open Systems Interconnection—File Transfer, Access and Management—Part 2: Virtual Filestore Definition.
- 3) ISO 8571-3, Information Processing Systems—Open Systems Interconnection—File Transfer, Access and Management—Part 3: File Service Definition.
- 4) ISO 8571-4, Information Processing Systems—Open Systems Interconnection—File Transfer, Access and Management—Part 4: File Protocol Specification.
- 5) 棟上昭男他: OSI の応用, 308 p., 日本規格協会 (1987).
- 6) FTAM 実装規約書, INTAP-S004, (財) 情報処理相互運用技術協会 (1987).
- 7) 齊藤忠夫他: OSI の実現とその課題 (I) OSI 機能標準化の動向, 情報処理, Vol. 29, No. 9, pp. 1023-1031 (1988).
- 8) MANUFACTURING AUTOMATION PROTOCOL SPECIFICATION VERSION 3.0 (1987.7).
- 9) 姉崎章博: ASN. 1 ハンドラの設計と実装, 情報処理, Vol. 88, No. 11, pp. 1-8 (1988).
- 10) 中川路哲男他: 国際標準に準拠したファイル転送プロトコルの実現と評価, 情報処理学会論文誌, Vol. 29, No. 11, pp. 1071-1078 (1988.11).

(昭和 63 年 11 月 22 日受付)