

# OSI プレゼンテーション, ACSE, FTAM プロトコルの 実装と評価†

小花 貞夫<sup>\*\*</sup> 加藤 聰彦<sup>\*\*</sup> 鈴木 健二<sup>\*\*</sup>

筆者らは、OSI (開放型システム間相互接続) のプレゼンテーション層, ACSE (アソシエーション制御サービス要素) および FTAM (ファイル転送, アクセスと管理) のプロトコルのソフトウェアを実装した。本ソフトウェアでは、OSI 上位層, 特に今後多様化する応用層の実装に対して柔軟に対応できる汎用的なソフトウェア構成ならびに FTAM と実ファイル・システムとの実用的な対応付けを実現している。本論文では、OSI 上位層の実装手法, 特に複雑化, 多様化する応用を効率よく実現するソフトウェア構成, 抽象構文/転送構文の取扱い, 実システムとの対応付けなどについて論じ, 次いでこの結果に基づいて実装したプレゼンテーション, ACSE および FTAM のソフトウェアについて詳細に論じる。さらに実証実験等を通して, ソフトウェアの機能, 効率等について評価, 考察する。

## 1. はじめに

現在 ISO, CCITT 等では, 異機種 of 計算機システムあるいは端末間で通信を可能とするための, OSI (開放型システム間相互接続) の標準化を精力的に進めている。これまでにトランスポート層, セッション層をはじめ, プレゼンテーション層ならびに応用層における ACSE (アソシエーション制御サービス要素) や, FTAM (ファイル転送, アクセスと管理)<sup>1)</sup> などの特定の応用サービス要素のサービス定義およびプロトコル仕様が IS (国際標準) となっている。これにより OSI のすべての層に関する標準がそろいはじめたこととなる。今後は, これらの標準に準拠したシステムをいかに実現していくかが重要なポイントとなる。

OSI 実装に関する手法については, これまでにもいくつか論じられている<sup>2)-6)</sup>。しかしこれらは OSI の 7 層全般に共通する基本的な手法に関するもの<sup>2)</sup>, あるいはセッション層以下に関するもの<sup>4)</sup> であったり, またプレゼンテーション以上の上位層を扱っていても単一 of アプリケーション (応用) しか動作しない環境を想定しているもの<sup>6)</sup> であり, 複数の異なる応用が同時に動作する上位層の効率よい実現方法を扱ったものではない。

本論文では, 複数の異なる応用が同時に動作できる上位層の環境を実現するという観点より, ソフトウ

アの構成, 抽象構文/転送構文の取扱い, 実システム環境との対応などの応用層に特有な問題について論じるとともに, その結果に基づいて開発したプレゼンテーション, ACSE および FTAM のソフトウェアの詳細について論じる。さらに実証実験等を通して, これらのソフトウェアの機能, 効率等について評価, 考察する。

## 2. プレゼンテーション層, ACSE, FTAM の概要

まずプレゼンテーション層と応用層, 特に ACSE, FTAM の機能の概要について述べる (図 1 参照)。

### 2.1 プレゼンテーション層

OSI では応用プロセス間で通信する際に扱われる情報を, 抽象構文 (Abstract Syntax) と転送構文 (Transfer Syntax) という 2 つの側面からとらえている。抽象構文はデータの意味・解釈を与えるための構文で, 応用層で取り扱われる。一方転送構文は転送上の表現形式 (符号化形式) を与えるもので, プレゼンテーション層で取り扱われる。これら抽象構文と転送構文を表現する手法として, ASN. 1 (抽象構文記法 1) とその基本符号化規則<sup>2)</sup> が別標準としてそれぞれ規定されている。プレゼンテーション層では, 抽象構文と転送構文の関連付け (プレゼンテーション・コンテキストと呼ばれる) を制御する機能と抽象構文/転送構文間の相互変換を行う機能を持つ。つまり応用層からの要求に応じて, 抽象構文に対応する適切な転送構文を決定し, これに従ってデータを転送する。プレゼンテーション・コンテキストは, プレゼンテーション・コネクション確立 (P-CONNECT) 時, あるいは

† Implementation of OSI Presentation, ACSE and FTAM Protocol Softwares, and Its Evaluation by SADA OBARA, TOSHIHIKO KATO and KENJI SUZUKI (KDD Kamifukuoka R & D Laboratories).

\*\* 国際電信電話 (株) 上福岡研究所

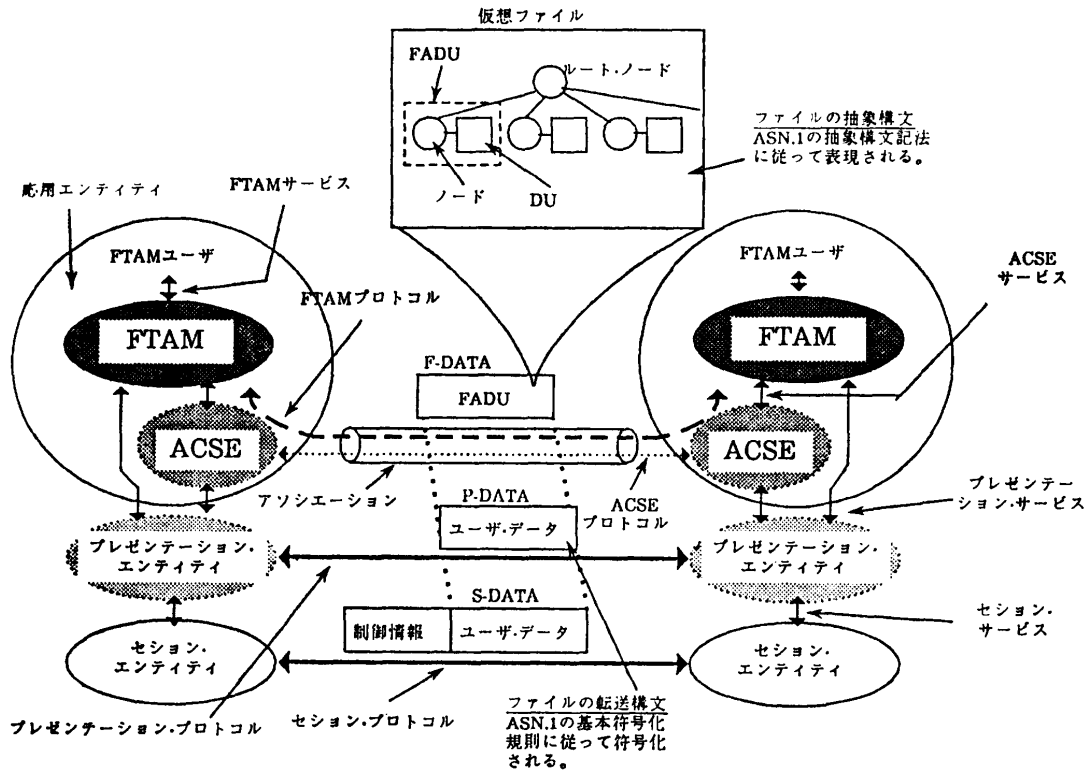


図1 プレゼンテーション, ACSE および FTAM の関係  
 Fig. 1 Relationship among presentation, ACSE and FTAM.

プレゼンテーション・コンテキスト変更 (P-ALTER-CONTEXT) 時に、相手プレゼンテーション・エンティティとの間で折衝し設定される。

プレゼンテーション層はさらに応用層がセッション層のアクティビティ、トークン、同期などの会話制御機能を直接使用できるようなインタフェース (P-ACTIVITY-START, P-TOKEN-GIVE, P-SYNC-MINOR など) 機能も提供している。

2.2 ACSE

応用層は、抽象構文という形でデータを授受して、特定の通信業務を遂行する。応用層の機能を遂行する実体である AE (応用エンティティ) は、いくつかの ASE (応用サービス要素) から構成される。ACSE は応用層のひとつの ASE であり、すべての通信業務 (応用) に共通な機能である応用アソシエーション (以下単にアソシエーションと呼ぶ) を制御する。アソシエーションは、双方の応用エンティティ間で情報を授受するための論理的な関連であり、ACSE により確立 (A-ASSOCIATE)、解放 (A-RELEASE, A-ABORT, A-P-ABORT) される。アソシエーションを確立する際には、AE の名称 (AE タイトルと呼ぶ) により相

手 AE を指定し、また応用コンテキストと呼ばれる応用層の通信環境 (使用する ASE やそれらの使用規則、オプションなど) を折衝し設定する。

2.3 FTAM

FTAM も、応用層のひとつの ASE で、ファイル転送という応用特有な機能を提供する。ここでは、遠隔にある異種のファイル・システム (仮想ファイル・ストアと呼ぶ) 間で、共通の手順によりファイル全体あるいは一部を転送したり、新たなファイルを生成したり、削除したりするためのプロトコルを規定している。

FTAM では、仮想ファイル・ストアの機能を提供する側 (レスポンド) とそれを利用する側 (イニシエータ) とが明確に区別されており、その動作は双方で不平衡となっている。

FTAM では、異なる計算機のさまざまな形態のファイルを扱うために、次のような仮想ファイルを定義している。

- 仮想ファイルは現実のファイル編成を、木構造により一般化して表現している。木の各ノードにはその名前とファイルの内容である DU (データ単位) が

付加され、ファイルの転送およびアクセスは木または部分木単位（このアクセス単位を FADU と呼ぶ）で行われる。さらに現実のファイル編成に対応させるため、木の構造、DU の内容や許される DU への操作等を制限したドキュメント・タイプが定義されている。ドキュメント・タイプには、無構造（ルートノードのみ）、シーケンシャル・フラット（木の深さが1でノード名はなし）等がある。

- 仮想ファイルは、ファイル名、ファイルに許される操作、操作を許可するユーザ名、ドキュメント・タイプを示すコンテンツ・タイプ、生成ユーザ名、生成日時などの属性を持つ。

また FTAM の手順では、まず FTAM 通信の開始 (F-INITIALIZE) を宣言し、次にファイルを選択 (F-SELECT) または生成 (F-CREATE) する。さらに選択または生成されたファイルをオープンし (F-OPEN)、データを送受信 (F-READ/F-WRITE, F-DATA など) したり、属性の参照 (F-READ-ATTRIB) や変更 (F-CHANGE-ATTRIB) を行う。

### 3. OSI 上位層プロトコル実装のための基本検討

プレゼンテーション層および应用層のプロトコルを実装する場合に重要となる以下の項目について基本的な検討を行った。

#### (1) ソフトウェア・プロセス構成

OSI 上位層プロトコルのためのソフトウェアの構成方法としては、次の2つの方式が考えられる。

- ① プレゼンテーション層以上の機能すべてを単一のプロセスとして構成する。
- ② プレゼンテーション層、ACSE や FTAM などの应用サービス要素を独立のマルチ・プロセスとして構成する。

①の単一プロセス方式では、②のマルチ・プロセスの方式に比べて、プロセス間インタラクションによるオーバーヘッドが存在しないものの、以下のような問題がある。

- 単一の应用（例えば、ファイル転送のみ）しか同時にサポートしない場合には問題とはならないが、ファイル転送 (FTAM)、メッセージ通信処理 (MHS)、ディレクトリ・システム、遠隔データベース・アク

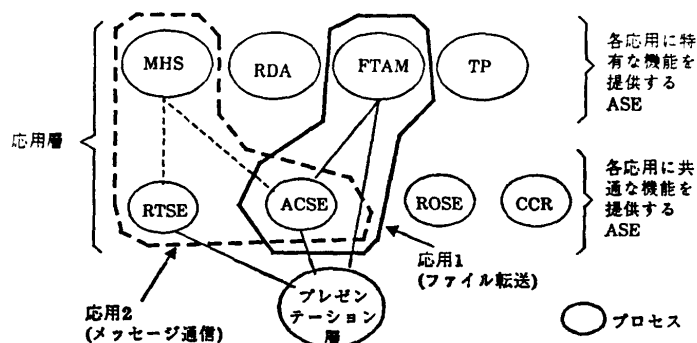


図 2 柔軟なプロセス構成

Fig. 2 Flexible software structure.

セス (RDA) やトランザクション処理 (TP) などといった複数の異なる应用を同時にサポートし、それらを複数ユーザから利用可能にしようとする、各应用ごとにプレゼンテーションや ACSE の機能モジュールを重複して持たせることとなり、実行時のプログラムの共有ができない。

- また複数の異なる应用を含めた形で全体を1つのプロセスとすることも考えられるが、全体のプロセス・サイズが非常に大きくなり、このためメモリのオーバーレイやスワッピングが頻繁に行われ、効率低下が生ずる可能性がある。
- ある機能モジュール（例えば、FTAM の機能モジュール）で、実ファイルなどの I/O 待ちになった場合、プロセス全体が I/O 待ちとなるので、全体としてのサービス性が低下する可能性がある。

これに対して、②のマルチ・プロセス方式では、プロセス間インタラクションによるオーバーヘッドが多少あると考えられるが、上記のような問題はなく、特に図2に示すように各種 ASE の機能モジュールを組み合せることにより、複数の異なる应用を同時にサポートする環境を効率よく実現できる。

#### (2) 抽象構文/転送構文の取扱い

2.1 節で述べたように抽象構文および転送構文は、プレゼンテーション層と应用層に特有なデータ表現であり、OSI の上位層の実装においては、それらをいかに取り扱うかが重要となる。

抽象構文は、実プロセスが扱う実システム固有のデータ構造を OSI 環境で共通に理解できるデータ構造にモデル化したものである。したがってプレゼンテーション層における抽象構文と転送構文の変換では、直接実システムのデータ構造（例えばファイル構造や操作コマンド）と転送構文とを対応付けるという

方法が考えられる。つまり FTAM の場合、ファイル・データを実ファイル・システムの書き込みコマンドの引数に直接対応付けることになる。

しかしながらこのような方法を採用すると、効率的な抽象構文／転送構文解析が行えるものの、以下のような問題が生じる。

- ACSE や FTAM などの ASE でプロトコル・エラーが生ずるような応用層プロトコル・データ単位 (APDU) の場合にも、プレゼンテーションで APDU の意味解釈をも含めた不必要な構文変換処理を行ってしまう。
- 応用プロセス (あるいは ASE) による FTAM の使用形態は画一的ではなく、例えば、ある応用では直接ローカル・ファイル・システムに書き込み、またある応用では受信ファイル・データを応用プログラムが直接利用するなどの場合に柔軟に対応できない。
- 3章(1)で述べたようなマルチ・プロセス構成において、プロセス間で構造化された実システムのデータ構造を授受させるのは難しい。

このため、プレゼンテーション層と応用層のプロセス間のインタラクションでは、抽象構文として特定の応用や特定のシステムに依存しない内部表現を用いることが重要である。

図3に示すように、プロセス間インタフェースのユーザ・データにおける抽象構文に対する内部表現としては、以下の理由によりプレゼンテーション層の ASN.1 基本符号化規則による転送構文を採用することとした。

- OSI 応用層の各種プロトコルやデータを規定するために現在使用されている ASN.1 記法による抽象構文と極めて親和性が高いこと。
- 将来 ASN.1 の基本符号化規則以外の符号化規則が開発され、転送構文のひとつとして追加される場合、ならびにプレゼンテーション層におけるデータ圧縮や暗号化 (これらも別の転送構文として扱われている) を行う場合にも、実

際の転送構文と、内部表現との変換が容易であること。

(3) 実システム環境との対応

OSI ソフトウェアの実装で、OS (オペレーティング・システム) など実システム環境の影響を直接受けるものとして、一般に I/O 装置とのインタラクション、プロセス間のインタラクションなどがあげられるが、応用層における FTAM など各応用業務特有の ASE では、さらに実システムの応用そのもの (例えばファイル管理サービス) と対応をとることも重要である。ここでは、ファイル転送、ファイル・アクセスやファイル管理のための FTAM の場合を例として検討する。

FTAM の仮想ファイル・ストアを実現する方法としては、すでに実システムが提供するファイル管理機能を利用し、FTAM で規定される機能と実システムで提供される機能との間で、以下の項目について対応をとるのが一般的な手法と考えられる。

- ファイル操作  
ファイルのオープン、レコードの読み書き、ファイル属性の参照／設定など。

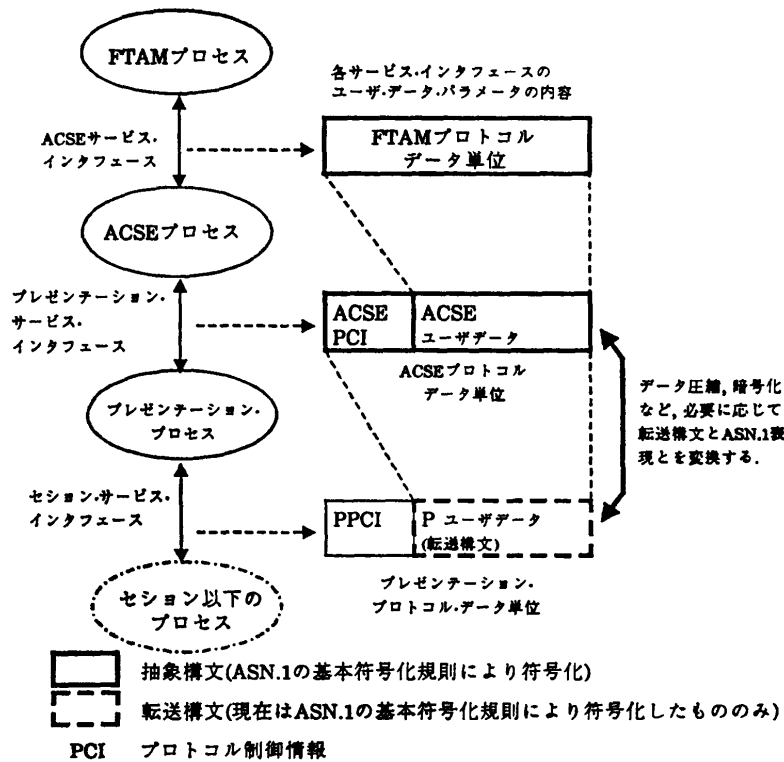


図3 抽象構文の内部表現  
Fig. 3 Internal representation of abstract syntax.

- ファイル構造  
ファイル編成など。
- ファイル属性  
ファイル名, 作成日時, ファイル・サイズなどの属性。
- アクセス制御  
パスワードによるユーザの認証や, ファイルに対するアクセス制御など。

しかしながら, FTAM で規定される機能で, 実システムで同様な機能を提供していないものをどのように対応付けるかの基準が重要となる。例えば, FTAM では多くのファイル属性を扱えるようにしているが, この中には実際のファイル・システムではサポートされていないものも存在する。

OSI では, 実システムとの対応に, どのような基準を設けるかについては実装上の問題として扱っている。そこで, ここでは実システム環境と OSI 環境の両環境でのファイルの共通利用を可能とする観点より, 対応付けの基準を検討する。

基本的に FTAM の標準で必須とされる機能 (あるいはパラメータ) はサポートする必要がある。しかしながら必須でないオプション機能については, 以下の基準を設けることが肝要である。

ア) 対応付けられる場合には, 対応をとる。

イ) 実ファイル・システムにない情報 (例えば実ファイル・システムでサポートしない属性情報など) を, 実ファイル・システムとは独立な情報として新たに持つ必要がある場合には, 機密保護など運用上必須なものを除いて, 基本的にサポートしない。

イ)は実システム環境と OSI 環境の両環境でのファイルの共通利用という観点から, 例えば遠隔のシステムから FTAM を使って自システムにコピーしたファイルを, ローカルなユーザが利用する場合に, 通常のファイル・システム操作のみではファイルを正しく扱えないことを避けるためである。

#### 4. プレゼンテーション, ACSE および FTAM の実装

上記の OSI 上位層実装の検討結果に基づいて, 筆者らはプレゼンテーション, ACSE, FTAM プロトコルを実装した。

##### 4.1 実装の前提

ここでは, 以下の前提のもとに実装を行った。

- VAX 11/8700 (OS: VMS) 上に実装する。
- すでに実装しているトランスポート, セッション<sup>4)</sup>の上に, 新たにプレゼンテーション, ACSE および FTAMを実装し, 異機種システムとの相互接続が可能のように OSI に完全に準拠した形で実現する。
- 基本的にプレゼンテーション, ACSE および FTAM (DIS 版: 国際規格案版) のすべてのプロトコル要素を実現する。
- FTAM の仮想ファイルについては, OS が提供するファイル管理機能 (今回は VMS が提供する RMS: Record Management Service) と対応させることによって実現する。
- ソフトウェアの移植性を考慮し, C言語によりプログラミングする。

#### 4.2 実装範囲

今回の実装では, 以下の範囲を実装の対象とした。

- プレゼンテーション, ACSE および FTAM の機能で, 今回実装の対象とした機能を表 1 に示す。
- FTAM のイニシエータ (ファイル・ユーザ側) とレスポнда (仮想ファイル・ストア側) の両機能を実現する。

表 1 実装機能  
Table 1 Functions implemented.

FTAM	仮想ファイル・	<ul style="list-style-type: none"> <li>●ドキュメント・タイプ 無構造テキスト, 無構造バイナリ, シーケンシャル・フラット・テキスト, シーケンシャル・フラット・バイナリ</li> </ul>
	サービス/プロトコル	<ul style="list-style-type: none"> <li>●サービス・レベル User Correctable (UCFS)</li> <li>●サービス・クラス ファイル転送, ファイル管理 (一部)</li> <li>●機能単位 カーネル, Read, Write, Limited File Management, Enhanced File Management (一部), Grouping</li> <li>●プロトコル・タイプ Basic File Protocol, Basic Bulk Data Transfer Protocol</li> </ul>
ACSE		<ul style="list-style-type: none"> <li>●カーネル機能単位 アソシエーション制御</li> </ul>
プレゼンテーション		<ul style="list-style-type: none"> <li>●カーネル機能単位 コネクション制御, プレゼンテーション・コンテキスト制御, セッション・パススルー</li> </ul>

- 同時に複数の FTAM ユーザをサポートする。
- FTAM のイニシエータ側のユーザ・インタフェースとしては、OS のコマンド・レベルと応用プログラムからの関数呼び出しレベルをサポートする。

4.3 ソフトウェア構成

(1) ソフトウェアは、図4 (ソフトウェア・プロセス構成) および図5 (ソフトウェア構成の詳細) に示すようなソフトウェア・モジュール群から構成し、複数の異なる応用が同時に動作できるように、プレゼンテーション、ACSE, FTAM およびユーザ・プログラムは独立のプロセスとした。各プロセスは上り/下りのキューで結合される。今回、キュー管理には VMS のメールボックス機能を用いた。なおアソシエーションの確立/解放を伴う場合以外 (例えば F-DATA が直接 P-DATA にマッピングされるなど) は、図4中◆に示す通り、ACSE を介さず FTAM とプレゼンテーションが直接データ授受できるようなキューイングのパスを用意し、不必要なプロセス間インタラクションが行われないようにした。

(2) FTAM のイニシエータ側とレスポンド側とは、一方がファイル提供者側で、もう一方がその利用者側という FTAM 本来の使用形態から、以下のような不平衡な作りとした。

- イニシエータ側としては、今後の応用層構造の拡充 (他の ASE が FTAM を制御するなど) を考慮

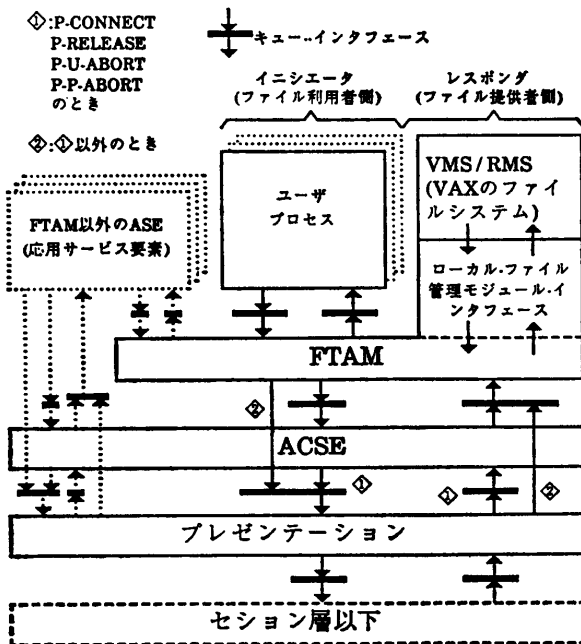


図4 ソフトウェアプロセス構成  
Fig. 4 Software process structure.

し、ユーザ・プログラムとユーザ代行プログラムを、FTAM プロセスとは独立なユーザ・プロセスとして用意した。ユーザ・プログラムとユーザ代行プログラムは単一のユーザ・プロセスで構成され、複数のユーザ・プロセス (異なる種類のものでよい) が FTAM プロセスの上で動作できる。

今回作成したユーザ代行プログラムは、ユーザが FTAM を使った各種の応用プログラム (ユーザ・プログラム) を作成しやすいように、関数呼び出しの形でファイル操作 (例えばファイル全体をコピーする) を行えるインタフェースを提供する。

- またレスポンド側としては、OS が提供するファイル管理システムと対応をとるローカル・ファイル管理モジュール・インタフェース・プログラムを FTAM のプロセスに組み込んだ。

(3) プレゼンテーション、ACSE および FTAM の各プロセスは、基本的に入出力管理、サービス・プリミティブ/PDU (プロトコル・データ単位) 解析、コネクション (あるいはアソシエーション) 管理、状態遷移制御、プロトコル生成の各プログラムから構成される。

入出力管理プログラムはメールボックスからのサービス・プリミティブの読み出し/書き込みを行う。サービス・プリミティブ/PDU 解析プログラムは、サービス・プリミティブ (あるいは PDU) の種類を解析する。コネクション (あるいはアソシエーション) 管理プログラムは、複数のコネクション (あるいはアソシエーション) の状態を管理し、状態遷移制御プログラムは、状態と発生したイベント (サービス・プリミティブあるいは PDU など) からとるべきアクションと状態遷移を決定し、実行する。この時サービス・プリミティブや PDU の生成が必要ならば、プロトコル生成プログラムが呼び出される。

このほか、プレゼンテーションや FTAM のプロセスには、転送構文チェック、ローカル・ファイル管理モジュール・インタフェースのプログラムが存在する。転送構文チェック・プログラムは、ACSE の PDU, FTAM の PDU および FADU (ファイル・データ) の転送構文上の正当性をチェックする。ローカル・ファイル管理モジュール・インタフェース・プログラムについては、4.4 節に述べる。

(4) FTAM プロセスは複数の異なる種類のユーザ (異なるユーザ代行プログラムや FTAM 以外の ASE) をサポート可能とするために、上位に提供する

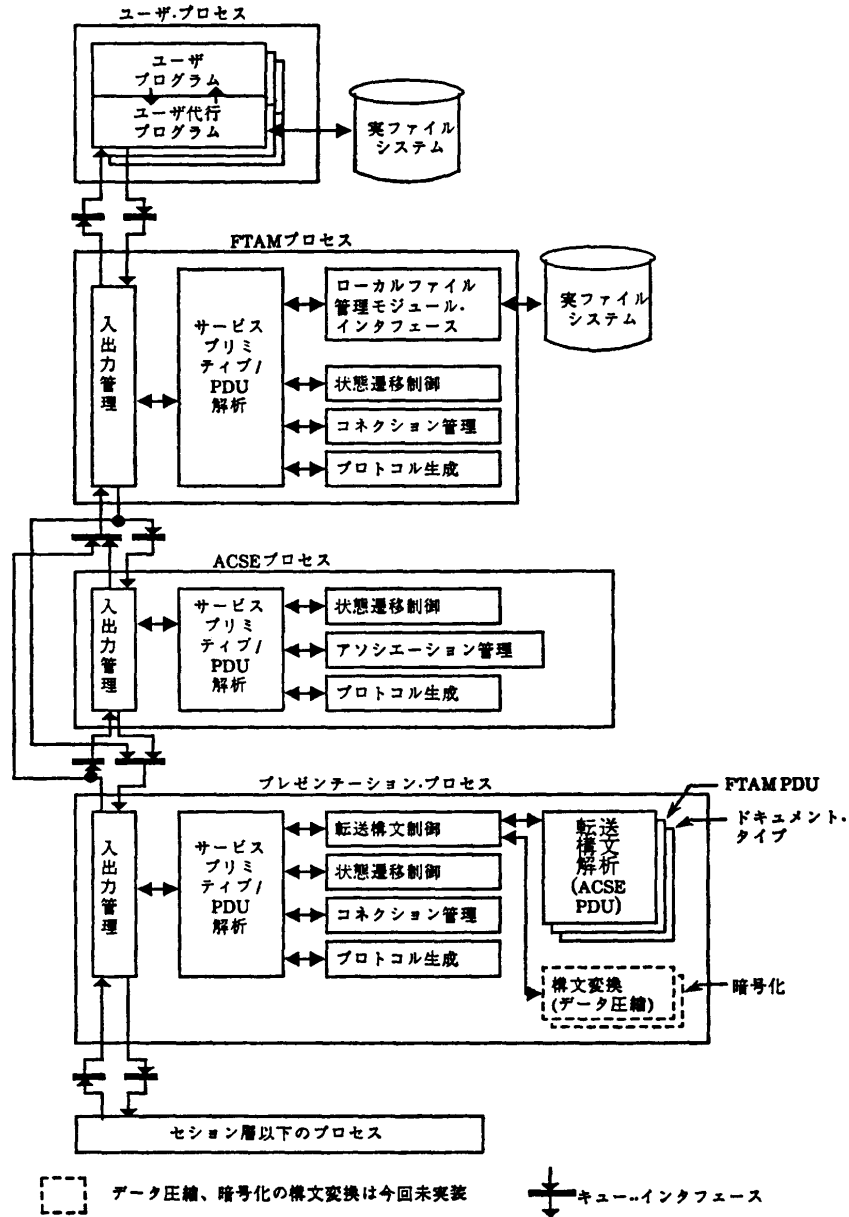


図 5 ソフトウェア構成の詳細  
Fig. 5 Detailed software structure.

キュー名 (ここではメール・ボックス名) と FCEP (FTAM Connection End Point) 番号を用いて個々の FTAM コネクションを管理させた。FCEP 番号は、ユーザ代行プログラムあるいは他の ASE プロセスの中で、ユニークに番号付けされ管理される。またアソシエーションの確立/解放を伴う場合以外は、ACSE を介さずに FTAM とプレゼンテーションが直接データ授受を行えるようにするため、[FTAM/ACSE 間の上り キュー名+AEP (Association End

Point) 番号] および PCEP (Presentation Connection End Point) 番号がプレゼンテーション・コネクションおよびアソシエーション確立時に、要求/確認あるいは指示/応答プリミティブを使って FTAM とプレゼンテーションが ACSE を経由して通知し合うようにした。ここで AEP 番号は ACSE がアソシエーションを管理、識別するためのもので、PCEP 番号はプレゼンテーション・プロセスがプレゼンテーション・コネクションを管理、識別するためのもので

ある。FCEP 番号と AEP 番号の概念は OSI 標準にはなく、今回の実装のために新たに導入した。

4.4 FTAM とローカル・ファイル・システムとの対応

ローカル・ファイル管理モジュール・インタフェース・プログラムは、レスポンド側の仮想ファイル・ストアを実現するため、実ファイル・システム（ここでは RMS）との間で以下のような機能の対応付けを行っている。

(1) ファイル操作の対応付け

実ファイル・システムでは、FTAM の操作とほぼ同様なシステム・サービスが提供されているため、比較的容易に対応がとれる。ファイルの書き込み手順における、今回の実装で行った FTAM と RMS のシステム・サービスとの対応付けを図 6 に示す。

なお FTAM の操作と実ファイル・システムのシステム・サービスとは、指定するパラメータの種類に多少差がみられる。例えば F-OPEN 要求のコンカレ

ンシ制御は F-SELECT 要求よりも制限の強い指定が可能であるのに対して、VMS では F-OPEN に対応するシステム・サービス (SYS\$CONNECT) でコンカレンシ制御を指定できない。このような場合、今回の実装ではファイルをいったんクローズし、新たなコンカレンシ制御に従って再度オープンすることで対応をとるなどの工夫が必要であった。

(2) ファイル構造の対応付け

実システムのファイルに対して、順/索引/相対などのファイル編成のほか、固定長/可変長/ストリーム等のレコード・フォーマット、CR (キャリッジ・リターン) 制御/印字書式制御等のレコード属性などの組合せにより、仮想ファイルの木構造と対応をとる。今回表 2 に示すような対応付けを検討し、現在までの実装では、無構造とシーケンシャル・フラットの対応を実現している。

ただし表 2 のシーケンシャル・フラットのように、いくつかの異なる実ファイルに対応する場合があります。

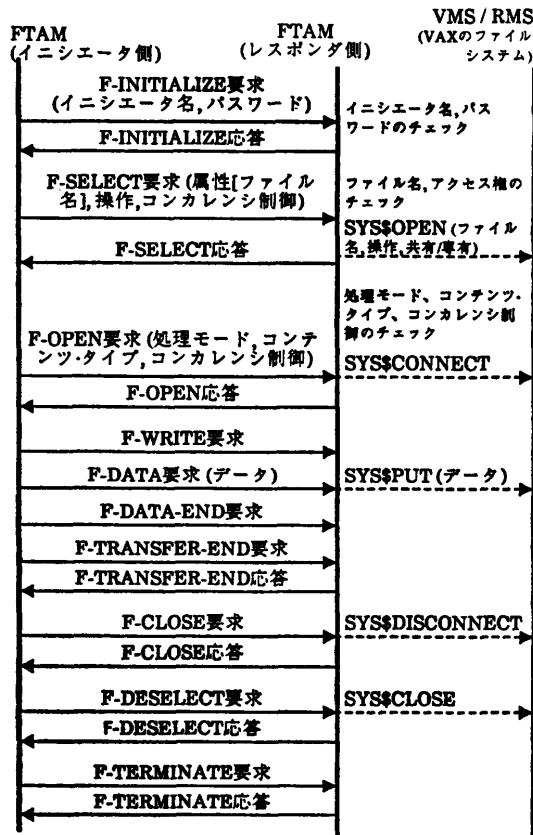


図 6 FTAM と実ファイル・サービス (RMS) とのファイル操作の対応例

Fig. 6 Correspondence on file operations between FTAM and local file service (RMS).

表 2 実ファイル (RMS) と仮想ファイルにおけるファイル構造の対応  
Table 2 Correspondence between local file (RMS) and virtual file.

実ファイル	仮想ファイル
編成: 順編成 レコードフォーマット: ストリーム レコード属性: CR 制御 (例 ユーザが C 言語の標準入出力で作成したファイル)	無構造
編成: 順編成 レコードフォーマット: 可変長 レコード属性: CR 制御 (例 エディタで作成したファイル)	シーケンシャル・フラット
編成: 順編成 レコードフォーマット: 可変長 レコード属性: なし (例 オブジェクト・ファイル)	シーケンシャル・フラット
編成: 順編成 レコードフォーマット: 固定長 レコード属性: なし (例 EXEC ファイル)	シーケンシャル・フラット
編成: 相対編成 レコードフォーマット: 固定長/可変長	ユニーク名付オーダード・フラット
編成: 索引編成 (主キーのみ/重複可) レコードフォーマット: 固定長/可変長	オーダード・フラット
編成: 索引編成 (主キーのみ/重複不可) レコードフォーマット: 固定長/可変長	ユニーク名付オーダード・フラット



FTAM の機能ではこれらを区別することができない。このような場合ファイル名の一部 (VMS ではファイル・タイプ, 例えば test. OBJ の "OBJ") によってどの実ファイルに対応付けるかを決定するなどの措置を講ずる必要があった。

### (3) ファイル属性の対応付け

仮想ファイルの必須の属性であるファイル名とコンテンツ・タイプについては以下のように対応付けた。一般に OS では, ファイル管理に階層的なディレクトリ構造を持っており, またバージョン管理も行っているが, FTAM ではこれらを指定する手段がないため, ファイル名の一部にディレクトリ名やバージョン番号の指定も許すことで対処した。またコンテンツ・タイプ (ドキュメント・タイプ) 中の DU の内容 (テキストかあるいはバイナリか) の区別については, 対応する情報を OS で管理しないことが多く, ファイル名の一部 (VMS ではファイル・タイプ) からテキスト・ファイルかバイナリ・ファイルかを決定する

こととした。その他のオプションの属性については, 最後にファイルを読んだユーザ名やその日時など VMS のファイル属性にないものは対応付けなかった。

### (4) アクセス制御の対応

F-INITIALIZE でのイニシエータ名のパスワードによるイニシエータの正当性の確認に対しては, OS でパスワードの検査機能がユーザに提供されていない場合が多く, このため OS とは別にパスワードを管理する必要がある。また仮想ファイルに許される操作は OS が管理するファイルのアクセス権に対応させるが, OS ではユーザの識別およびファイルのアクセス制御に利用者識別のためのコード (VMS では UIC) を用いている。このためイニシエータ名/パスワード/利用者識別コードの対応情報を別途管理する必要があった。

## 4.5 FTAM 関連プロセス

(1) FTAM の各サービス・プリミティブは表 3 に示すように下位の ACSE, プレゼンテーション,

表 3 FTAM, ACSE, プレゼンテーション, セッション・サービスの対応  
Table 3 Service mapping among FTAM, ACSE, presentation and session.

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-P-ABORT注1)	A-ABORT or A-P-ABORT注1)	P-U-ABORT or P-P-ABORT	S-U-ABORT or S-P-ABORT
F-U-ABORT注1)	A-ABORT注1)	P-U-ABORT	S-U-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-ATTRIB	→	P-DATA	S-DATA
F-CHANGE-ATTRIB	→	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注4)	→	P-DATA	S-DATA
F-LOCATE	→	P-DATA	S-DATA
F-ERASE	→	P-DATA	S-DATA
F-READ	→	P-DATA	S-DATA
F-WRITE	→	P-DATA	S-DATA
F-DATA注2)	→	P-DATA	S-DATA
F-DATA-END	→	P-DATA	S-DATA
F-TRANSFER-END	→	P-DATA	S-DATA
F-CANCEL注4)	→	P-RESYNCH (abandon) or P-DATA注3)	S-RESYNCH (abandon) or S-DATA
F-CHECK注2)注4)	→	P-SYNC-MINOR	S-SYNC-MINOR
F-RESTART注4)	→	P-RESYNCH (restart)	S-RESYNCH (restart)

注1) 今回 F-P-ABORT, F-U-ABORT, A-ABORT は PDU を実装せず, プレゼンテーションの異常終了を直接対応付けた。

注2) PDU はない。

注3) FTAM の Recovery 機能単位とプレゼンテーションの Resynchronize 機能単位を選択した場合には P-RESYNCH (abandon) に, それ以外は P-DATA にマッピングされる。

注4) 今回障害回復等のためのサービスは未実装。

セッションのサービス・プリミティブにマッピングさせた<sup>6)</sup>。

(2) ファイル構造として、無構造ファイルとシーケンシャル・フラット・ファイルをサポートした。無構造ファイルの場合には、一定長ごとに分割して F-DATA として送出し、またシーケンシャル・フラット・ファイルの場合には、各レコードを F-DATA として送出手続にした。

(3) (N)-アドレスと (N)-SAP ((N)-サービス・アクセス点) との対応は 1:1 とし、また AE タイトルとプレゼンテーション・アドレスとの対応は別ファイルとして対応表を管理した。

(4) 今回作成したユーザ・プロセス (ユーザ・プログラムとユーザ代行プログラム) では、自システムのファイルと遠隔システムとの間でファイルの複写を行うための機能を OS のコマンド形式 (例えば、"COPY ファイル名<sub>1</sub> TO ファイル名<sub>2</sub>") で呼び出せるようにした。また仮想ファイルのファイル属性の参照 ("READATTRIB ファイル名"), ファイルの削除 ("DELETE ファイル名") などの機能も実現した。さらに、FTAM の標準では規定されていないが、仮想ファイル・ストアのディレクトリ参照 ("DIR ディレクトリ名") を可能とした。ここではディレクトリ参照もひとつのシーケンシャル・フラット・ファイルの読み出しとして扱うことで実現した。

#### 4.6 ACSE, プレゼンテーション・プロセス

(1) ACSE プロセスは、標準で規定されるアンシエーション制御機能をすべてサポートした。またプレゼンテーション・プロセスは、コネクション制御、プレゼンテーション・コンテキスト制御 (現在 P-ALTER-CONTEXT は実装していない) およびセッション・パススルー機能などの諸機能をサポートした。

(2) FTAM PDU をプレゼンテーションで効率よく転送できるよう、サイズの小さい複数の FTAM PDU など (複数 F-DATA や F-BEGIN-GROUP/F-END-GROUP によるグルーピング) を連結してひとつの P-DATA にマッピングできるようにした。

(3) 現行セッション標準 (バージョン 1) におけるセッションの異常終了サービス (S-U-ABORT) のユーザ・データ長が 9 オクテットと限定されており、上位層の異常終了 (P-U-ABORT, A-ABORT, F-U-ABORT など) のプロトコル・データ単位 (PDU) を収容することができないという問題があるため、今

回異常終了についてはプレゼンテーションの PDU を実装し、ACSE と FTAM では、プレゼンテーションの異常終了サービスを ACSE と FTAM のそれぞれの異常終了サービスに直接対応させる必要があった。

## 5. 評価および考察

### 5.1 効率について

実装したプレゼンテーション、ACSE、FTAM のソフトウェアの処理効率を評価するために、以下の試験を行った。

ここでは、VAX 8700 上にトランスポート層以上のプロセス群を FTAM のイニシエータ側とレスポンド側とにそれぞれ別に用意し、①ネットワーク層を VAX 8700 上で内部折返しする、②ネットワーク層を X.25 公衆パケット網で外部折返しするという 2 つの形態で評価試験を行った。またここでは、トランスポートにはプロトコル・クラス 0 を、セッションには BCS (全二重) を使用した。

#### 1) ネットワーク層の内部折返し試験

ファイル長が 105 K バイトの無構造ファイルとシーケンシャル・フラット・ファイルを使って、それぞれの場合の転送効率を測定した。

表 4 は、イニシエータ側からレスポンド側 (仮想ファイル・ストア側) にファイルを書き込んだ (F-WRITE) 時の、ファイル転送効率およびレスポンド側における各プロセスの CPU 実行時間等を示している。ここで転送時間には、ファイル選択 (F-SELECT) からファイル解放 (F-DESELECT) までの時

表 4 転送効率とプロセス CPU 実行時間  
Table 4 Throughput and CPU time for each process.

プロセス	ファイルの種類	
	無構造ファイル (F-DATA 長 1,024 バイト) (秒)	シーケンシャル・フ ラット (F-DATA 長 80 バイト) (秒)
FTAM プロセス	3.0	3.4
ACSE プロセス	0	0
プレゼンテーシ ョン・プロセス	1.0	1.6
セッション層以下の プロセス	4.0	4.1
転送時間 (転送効率)	15 (56 Kbps)	17 (49.6 Kbps)

注) 環境: ネットワーク層内部折返し, 転送ファイル・サイズ 105 K バイト, セッション BCS (全二重), トランスポート・プロトコル クラス 0, TPDU 長 512 バイト

間が含まれる。したがって、この間 ACSE は関与しないため、ACSE プロセスの処理時間はゼロである。

表 4 に示す通り、転送効率として無構造ファイル転送の場合で、約 7 Kバイト/秒 (56 Kbps)、またシーケンシャル・フラット・ファイルの場合で約 6.2 Kバイト/秒 (49.6 Kbps) のスループットを確認できた。

またシーケンシャル・フラット・ファイルの場合、無構造ファイルより転送効率がやや落ちている。これは、主として以下の理由により、FTAM およびプレゼンテーションにおける F-DATA に対する ASN.1 の構文解析に必要となる処理が増加したこと起因する。

- 無構造ファイルでは、ファイル全体がひとつの FADU であり、転送の際にはこれを任意の大きさに分割し、それぞれを F-DATA として転送できる。したがって F-DATA 長を比較的大きく (実験では 1,024 バイト) 設定できる。これに対し、シーケンシャル・フラット・ファイルでは、ひとつの FADU が、例えばエディタで作成した順編成のテキスト・ファイルの 1 レコードに対応する。このため FADU を転送する F-DATA の長さが比較的小さく (実験では、80 バイト以下)、無構造ファイルと同じ規模のファイルを転送するためには、F-DATA 数がかなり増加する。
- さらにシーケンシャル・フラット・ファイルは、無構造ファイルとは異なり、FADU のレコード内容 (DU) のみでなく、ファイルのノード情報 (ノード記述子) も各レコードの間に挿入され、これも F-DATA として扱われるため、F-DATA 数をさらに増加させることとなる。

しかしながら、シーケンシャル・フラット・ファイルの場合であっても、レコード長が大きければ (例えば、プログラムの実行モジュール・ファイルでは、512 バイト)、無構造ファイルとほぼ同様のスループットを確認できた。

## 2) X. 25 公衆パケット網経由外部折返し試験

回線速度 9.6 Kbps、パケット長 128 バイトの X. 25 回線を 2 本 (イニシエータ側とレスポンド側) 使用し、パケット交換機で折り返して、試験を行った。この結果無構造ファイル、シーケンシャル・フラット・ファイルともに約 7.6 Kbps の実行スループットを確認できた。これは、9.6 Kbps の回線スピード・ネックに達していると考えられる。

以上の結果から、X. 25 (9.6 Kbps) の実回線では、

実装したソフトウェアが十分実用できることを確認できた。

また今回、LAN などの高速通信環境を実際に用いた効率測定は行えなかったが、かわりにネットワーク層の内部折返し試験を行うことで、ほぼそれに近い通信環境を設定し、評価を行った。この結果、1 台の計算機 (VAX 8700) 上でイニシエータ側とレスポンド側のプロセスを別に動作させ、擬似的に 2 つのシステムを実現するという通常とは異なる過負荷の環境下で評価したが、LAN などの高速通信環境においても効率面で実用できることを実証できた。ただし内部折返し試験での効率は計算機の性能に大きく依存するが、今回使用した VAX 8700 (6 MIPS) は、現在の計算機のレベルから考えてそれほど高速なものではないことを考慮すると、実装したソフトウェアのスループットは十分実用できると言える。

## 5.2 機能について

今回のプレゼンテーション層、ACSE および FTAM の実装を通して、以下のことが判明した。

### (1) ソフトウェア・プロセス構成

プレゼンテーションおよび応用層の各種 ASE の機能モジュールに対応するソフトウェアを独立なプロセスとしたマルチ・プロセス・ソフトウェア構成を採用することにより、ファイル転送 (FTAM)、メッセージ通信処理 (MHS)、ディレクトリ・システム、遠隔データベース・アクセス (RDA) やトランザクション処理 (TP) などといった複数の異なる応用を同時にサポートできる環境を実現できた。

また現在 OSI の応用層構造では、分散処理機能も考慮した柔軟な応用を実現可能とするため、アソシエーションごとの通信環境を制御する SACF (単一アソシエーション制御機能) や複数アソシエーションにまたがる通信環境を制御する MACF (複数アソシエーション制御機能) 等の検討も行っており、応用層の構造がより複雑化しつつある。例えば、ひとつの AE: 応用エンティティ (正確には、AEI: AE インボケーション) が複数のアソシエーションを扱い、しかも各アソシエーションごとに異なる ASE の組合せで通信が行えるようになっている。またアソシエーション上の ASE の組合せも通信中に動的に変更できる。このような応用層の構造に対しても、今回のマルチ・プロセスによる構成を使用すれば、MACF や SACF などのプロセスを追加することでスムーズに移行できる<sup>12)</sup>。

## (2) 抽象構文/転送構文の取扱い

プレゼンテーション層以上のプロセス間インタフェースにおける抽象構文の内部表現として、ASN. 1の基本符号化規則による転送構文を採用することにより、各プロセスのプログラムの汎用性、移植性を高めるとともに、今後のプレゼンテーション層におけるデータ圧縮や暗号化のための転送構文にも容易に対応可能とすることができた。なおデータ圧縮や暗号化のための転送構文については、今後の標準化動向にあわせて実現する予定である。

## (3) 実システムとの対応

FTAM のファイル操作、ファイル編成、ファイル属性およびアクセス制御機能についての実ファイル・システムとの対応付けが実用的なレベルでほぼ可能であることを実証できた。ただし以下の点で、今後とも十分な検討が必要である。

- 今回解釈が比較的明確でどの計算機システムでも対応のとりやすい無構造ファイルとシーケンシャル・フラット・ファイルをサポートしたが、これ以外のファイル構造については、現在の FTAM で規定されている一般木構造と実ファイル構造との対応に不明確な点が多い（例えば、索引順編成ファイルの場合の主キー、副キーを FTAM の木構造でどのように対応させるかなど）。
- 一般に、実ファイル・システムでは、DU の内容（テキスト/バイナリ）を区別する手段がないため、ファイル名の一部（例えば VMS のファイル・タイプなど）を使って区別する等の手段を講ずる必要がある。
- 実際のローカル・ファイル・システムでは、一般にファイル編成のみでなく、レコードのフォーマットやレコードの属性なども含めてファイル構造を管理しており、DIS 版 FTAM の仮想ファイルではこれらが十分に対応付けられる機能を持っていない。このためこれらを考慮した FTAM の機能拡充が望まれる。

## (4) ディレクトリ参照機能と CCR の取扱い

- ファイル・ディレクトリのためのシーケンシャル・フラット・ファイルがあるかのように扱うことにより、遠隔の仮想ファイル・ストアのディレクトリ参照も可能とした。これは、現在の FTAM では規定されていないが、実際のファイル転送では、あらかじめ遠隔システムにあるファイル名やディレクトリ名を把握することは困難であるため、この機能が

実用上極めて有効なものである。FTAM プロトコルの機能拡充が望まれる。

- 今回、CCR（コミットメント、並行性と回復制御）など一部のオプション機能については、標準化の技術的内容に不十分なところがあるため、今後の動向に従って実装していくこととした。

## (5) IS（国際標準）版への移行

今回プレゼンテーション、ACSE および FTAM は、DIS 版標準に従って実装したが、IS 版標準への移行も容易である。特に FTAM では、今回未実装の F-RECOVER などの障害回復手順を含む外部ファイル・サービス（高信頼ファイル・サービス）機能等を追加実装することにより対応できる<sup>13)</sup>。

## 6. おわりに

本論文では、OSI（開放型システム間相互接続）のプレゼンテーション層、ACSE（アソシエーション制御サービス要素）および FTAM（ファイル転送、アクセスと管理）プロトコルのソフトウェア開発を通じ、OSI 上位層のソフトウェアの体系的な開発指針を示した。特に、今後複雑化する応用層の実装に対して柔軟に対応するためには、複数の応用を同時にサポートするためのマルチ・プロセス・ソフトウェア構成の適用、抽象構文/転送構文の取扱い、実システムとの対応付けが重要であることを論じた。さらに、実装したプレゼンテーション、ACSE および FTAM ソフトウェアを評価、考察し、本手法の有効性を示した。

今後とも、IS 版標準への対応や扱えるファイル形式（オーダード・フラットなど）の拡充など FTAM 機能の充実を行うとともに、他機種システムとの相互接続試験を行う予定である。また、本手法に基づき、ディレクトリ・システム、RDA（遠隔データベース・アクセス）や TP（トランザクション処理）などの応用プロトコルのソフトウェア開発も行う予定である。

謝辞 日頃御指導頂く KDD 上福岡研究所小野所長、浦野次長に感謝します。

## 参 考 文 献

- 1) ISO 8822, 8823, 8649, 8650, 8571/1-4.
- 2) ISO 8824, 8825.
- 3) Bochmann, G. V., Gerber, G. W. and Serre, J. M.: Semiautomatic Implementation of Communication Protocols, *IEEE Trans. Softw. Eng.*, Vol. SE-13, No. 9, pp. 989-1000 (1987).
- 4) 鈴木, 加藤, 浦野: OSI トランスポートおよびセッション・プロトコルの実装, 情報処理学会論文

- 誌, Vol. 29, No. 12, pp. 1180-1192 (1988).
- 5) 中川路, 勝山, 茶川, 水野: 国際標準仕様に準拠したファイル転送プロトコルの実装と評価, 情報処理学会論文誌, Vol. 29, No. 11, pp. 1071-1078 (1988).
  - 6) 小花, 鈴木, 加藤: OSI ファイル転送プロトコルとプレゼンテーション・レイヤ・サービスについての考察, 第 29 回情報処理学会全国大会論文集, pp. 933-934 (1985).
  - 7) 加藤, 鈴木: OSI FTAM 実装における抽象構文に関する一考察, 第 33 回情報処理学会全国大会論文集, pp. 1033-1034 (1986).
  - 8) 小花, 加藤, 鈴木: OSI FTAM 実装の基本設計, 第 34 回情報処理学会全国大会論文集, pp. 695-696 (1987).
  - 9) 加藤, 小花, 鈴木: OSI FTAM とローカルファイルシステムとの対応関係, 第 34 回情報処理学会全国大会論文集, pp. 697-698 (1986).
  - 10) 小花, 加藤, 鈴木: OSI FTAM, ACSE, プレゼンテーション・プロトコルの実装, 情報処理学会マルチメディア通信と分散処理研究会資料, 33-6 (1987).
  - 11) 小花, 加藤, 鈴木: OSI プレゼンテーション, ACSE および FTAM の実装と評価, 第 35 回情報処理学会全国大会論文集, pp. 521-522 (1987).
  - 12) 小花, 西山, 杉山, 鈴木: OSI における応用層構造 (ALS) の実現方法の提案, 第 37 回情報処理学会全国大会論文集, pp. 466-467 (1988).
  - 13) 小花, 加藤, 鈴木: OSI FTAM (ファイル転送, アクセスと管理) プロトコル・ソフトウェアの機能拡充, 第 36 回情報処理学会全国大会論文集, pp. 623-624 (1988).

(昭和 63 年 8 月 15 日受付)  
(平成 元年 4 月 11 日採録)



小花 貞夫 (正会員)

昭和 28 年生。昭和 51 年慶応義塾大学工学部電気工学科卒業。昭和 53 年同大学院修士課程修了。同年国際電信電話(株)入社。現在、同社上福岡研究所コンピュータ通信研究室主査。この間、パケット交換方式、ネットワークアーキテクチャ、OSI プロトコルの実装、データベース、国際ビデオテックス通信、分散処理技術の研究に従事。電子情報通信学会会員。



加藤 聰彦 (正会員)

昭和 31 年生。昭和 53 年東京大学工学部電気工学科卒業。昭和 58 年同大学院博士課程修了。工学博士。同年国際電信電話(株)入社。現在、同社上福岡研究所コンピュータ通信研究室主査。昭和 62 年から 63 年まで米国カーネギーメロン大学計算機科学科客員研究科学者。この間、OSI プロトコルの実装やコンフォーマンス試験、通信プロトコルの形式記述技法、分散オペレーティングシステムなどの研究に従事。昭和 60 年情報処理学会学術奨励賞受賞。電子情報通信学会会員。



鈴木 健二 (正会員)

昭和 20 年生。昭和 44 年早稲田大学理工学部電気通信学科卒業。昭和 44 年から 45 年までオランダのフィリップス国際工科大学に招待留学。昭和 51 年早稲田大学大学院博士課程修了。工学博士。同年国際電信電話(株)入社。現在、同社上福岡研究所コンピュータ通信研究室長。この間、磁気記録、パケット交換方式、ネットワーク・アーキテクチャ、通信処理などの研究に従事。昭和 62 年度、前島賞受賞。電子情報通信学会、IEEE 各会員。