国際標準仕様に準拠したファイル転送プロトコルの実現と評価†

中川路 哲男†† 勝山 光太郎††
芥川 哲雄†† 水野 忠則††

情報通信システムの発展に伴い，異機種を相互に接続する機会が増大している．国際標準機関では異機種間相互接続のためにOSIが開発されており，参照モデルに従った各層のプロトコルが標準化されつつある．なかでも高位層プロトコルは，ようやく標準が規定された段階のためプログラム量，性能などが未知であり，論理的なモデルの実世界へのマッピングなどの実装上の問題もあるため，その実現性を例証する必要があった．我々は高位層プロトコルでも標準化の進んでいるファイル転送プロトコルFTAMおよびACSE（アソシエーション制御），PP（プレゼンテーションプロトコル）を実装し，実装時の問題点やプログラム量，性能を明らかにしてOSIの実用性を確認した．さらに，業界標準であるファイル転送プロトコルFTPと機能面と性能面での比較を行い，その特性を評価した．また，実装に際しては，オブジェクト指向言語を採用するなど開発環境に工夫を凝らし，効率的な開発と，移植性，汎用性に富んだプログラムの作成を可能とした．

1. はじめに

情報通信の発展に伴い，異機種を相互に接続する機会が増えてきている．相互接続を円滑に行うため，ISOやCCITTにおいてOSI（開放型システム間相互接続）の開発が進められている．OSIの開発は基本参照モデルの制定に引き続き，各層のサービスとプロトコルの標準化が下位層から順次行われつつある．現時点では，第5層（セション層）までの規格化が完了し，第6層（プレゼンテーション層）および第7層（応用層）の規格化が進行中であり，実装が試みられる段階を迎えている．

OSIは様々なシステム間の接続を目標としており，既存の接続方式に捉われず，世界の専門家が新たに規定したものである．その意味で，非常に汎用性の高いアーキテクチャとなっている反面，以下の点でその実用性が疑問視されているのも事実である．

- 多層化オーバヘッドに起因する性能低下
- 多層化によるプログラム量の増大に起因する小型機への実装可能性
- 抽象度の高いモデルの実現性

OSIの実装に関しては，セション層以下に関する報告の事例が存在する[1]が，プレゼンテーション層以上についてはISOの標準が決まりかけたところなので，未だ実装上の問題点に関する報告が無く，実際に製作を行うことによってその実用性を検証する必要があった．

このため，我々は応用層の中で最も基本的なFTAM[2]（ファイル転送，アクセスと管理）およびACSE[3]（アソシエーション制御）とプレゼンテーション層のPP[4]（プレゼンテーションプロトコル）を三菱電機エンジニアリングワークステーションME 1000シリーズ（以下ME 1000と略す）上に実装し，これらの問題点を検討して実現性を例証する研究を行った．さらに，ME 1000に，業界標準とも言える異機種間ファイル転送プロトコルであるTCP/IP上のFTP[5]を実装することによって機能および性能上の比較を行い，FTAMの特性について評価を行った．

本論文では，FTAMの実装方針，開発環境，実装範囲および実装方式について述べ，さらにFTPとの比較について報告する．

以下，第2章では実装方針と開発環境について述べ，第3章では実現したプロコトルの実装範囲について報告する．第4章ではそれらに基づく実装方式について詳述し，第5章では本FTAMに対する性能を評価する．第6章ではFTPと比較することによって，機能や性能について考察し，さらに第7章で本実装結果をまとめる．

2. 実装方針と開発環境

2.1 実装方針

FTAM, ACSEおよびPPを実装するに当たり，我々は以下の点を実装方針とした．

（1） ホスト計算機からワークステーションまで各種

† Implementation and Evaluation of File Transfer Protocol Based on Standard Specification by TETSUO NAKAKAWAJI, KOTARO KATSUYAMA, TETSUO AKUTAGAWA and TADANORI MIZUNO (Mitsubishi Electric Corp.).

†† 三菱電機(株)

計算機に容易に移植可能であること.

(2) 高性能であること.

(3) 開発および試験の効率向上を図ること.

(4) 今後の他プロトコル実装時への応用を考慮して汎用的な構成方法とすること.

2.2 開発環境

上記の実装方針から，我々は以下に述べるように開発環境を工夫して，プロトコルの実装を行った.

(1) オブジェクト指向言語 superC[6] の採用

オブジェクト指向言語で記述されたソフトウェアでは，モジュールの独立性が高まり，それらを部品化して再利用することが可能となる．このことにより，ソフトウェアを効率的に開発できるだけでなく，今後の各種プロトコル開発時への再利用も期待できる．また，superC の処理系はプリプロセッサとして実現されており，プリプロセッサの生成コードがC言語であること，およびプリプロセッサ自体がC言語で記述されていることから，superC で記述されたプログラムは移植性にも優れている．さらに，superC はC言語との混合記述や静的束縛（コンパイル時にメソッド探索を実行）の機能を提供しているので，通常のオブジェクト指向言語で懸念される性能への影響も最小限に抑えることができる.

また，通信システムの参照モデルによるモデル化は，オブジェクト指向的な発想に基づいて行われている．すなわち，システムを階層化して各層にそれぞれ機能を割り当てて各層の独立性を高め，層間にサービスプリミティブというメッセージを定義し，隣接層との相互動作を記述している．したがって，モデルと親和性の高いソフトウェア構成とすることが可能となる.

(2) セションシミュレータ

通信ソフトウェアの試験は，実際に複数の処理装置を接続して実施されるが，この方式では下位層通信の影響を受けやすく，設備的にも準備を要するため，試験の初期段階からこのような方式で行うのは開発効率上望ましくない．我々はセション層以下の通信を模擬するセションシミュレータ上で，プレゼンテーション層以上のソフトウェアを試験する試験環境を構築し，開発効率の向上を図った．セションシミュレータの提供する機能を**表1**に示す.

セションシミュレータの利用においては，マルチウィンドウ機能を用いることにより，一台の ME 1000 上で複数処理装置間の通信状況が把握でき，効率的な開発および試験が可能となった.

表1 セションシミュレータの提供する機能
Table 1 Session simulater facilities.

トレースログ機能
逐次／連続実行機能
異常シーケンス発生機能
パススルーサービス機能
機能単位の折衝機能

表2 実装範囲
Table 2 FTAM implementation.

FTAM	機能単位 カーネル，読み出し，書き込み，グルーピング サービスクラス ファイル転送クラス サービスレベル ユーザコレクタブル アクセスコンテクスト US (Unstructured Single Data Units) ドキュメントタイプ テキスト，バイナリ	
ACSE	全機能	
プレゼンテーション	カーネル機能単位	
セション	カーネル，全二重	
トランスポート	クラス0	
ネットワーク データリンク 物理	(LAN) INP LLC タイプ2 MAC (CSMA/CD)	(広域網) X. 25 HDLC

3. 実装範囲

今回実現した各プロトコルの実装範囲を**表2**に示す.

FTAM の実装範囲は，最も利用度の高い，構造を持たない非構造型のファイルの全体転送に限定している．転送に当たっては，ファイルの内容や構造情報を通信相手と折衝する必要がある．折衝の方法として，ドキュメントタイプによる方法とコンストレインセットと抽象構文の組み合わせによる方法の2つがある．今回は転送情報量の少ない前者の方法を採用し，テキ

ストとバイナリの2つのドキュメントタイプを用意した．

プレゼンテーション層については，応用プロトコルとしてFTAMのみを想定したため，コンテクスト制御関連の機能単位は不要のため実装せず，カーネル機能単位のみを実装した．セション層以下の実装に関しては，開発途上段階ではセションシミュレータを利用し，実試験時にはME 1000のセション層以下の各プロトコルモジュールを使用した．フロー制御や順序制御などのデータの高信頼転送を保証する機能についてはトランスポートプロトコルのクラス選択によっても実現できるが，これらの機能は低位の層で実現した方が性能面において望ましいため，データリンク層でLLC (Logical Link Control) のタイプ2を選択することにより実現した．

4. 実装方式

OSIのプロトコルはその規定に当たって，抽象的なモデルを設定することで，規定内容が実装に依存することを避けている．したがって実装に当たっては，抽象的なモデルを実世界にマッピングするための各種の問題点が存在する．我々が直面した大きな問題点は以下の2点である．

- 抽象構文の実現方法
- 仮想ファイルストアと実ファイルシステムとのマッピング

本章ではそれらの問題点を挙げ，本実装におけるその解決および最終的なソフトウェア構成とプロセス構成について述べる．

4.1 抽象構文の実現

OSI基本参照モデルでは，データの表現形式の差を意識せずに応用エンティティ同士が通信できるように，プレゼンテーション層以下が扱うデータを転送構文として，応用層が扱うデータを抽象構文として明確に区別している．抽象構文を記述するための方法としてASN. 1[7]が標準化されており，応用層でのプロトコルデータは，ASN. 1で定義されている．両構文間の変換はプレゼンテーション層が行う．

抽象構文はデータをセマンティクスの側面のみから表現したものであり，これをプログラム上で実現する必要がある．その方法には，転送構文を採用する方法と，抽象構文を具体的な形に変換して表現する方法の2案が考えられる．前者の方法は，インタフェースは明確になるが本来プレゼンテーション層で行われるべき符号化／復号化処理が応用層とプレゼンテーション層に分散されるため，復号時のエラーが応用層内で発生するなどモデルと合致しない事象が発生し，標準に規定されていない処理を行う必要が生ずる．

今回は，汎用的なモデルに忠実な実装構造を実現することを重視して，抽象構文をC言語の構造体で表現する後者の方法を採用した．その実現に当たっては，ASN. 1で記述されたデータ定義をC言語の構造体に変換するプリプロセッサと，抽象構文に対する値の設定や獲得および転送構文との変換（符号／復号）を行うライブラリから成るツールを開発した．本ツールの構成を図1に示す．

本ツールの使用により，応用層のモジュールはデータを抽象構文の形で扱えるため，規格のデータ定義をそのままの形で扱うことができ，応用層プロトコルの実装負荷が軽減された．なお，本ツールはFTAMのみならず，他の応用層プロトコル開発時にも再利用可能である．

4.2 仮想ファイルの実現

FTAMでは，各システム間のファイル管理方式の差を吸収するため，図2に示すように仮想ファイルシステムという形でファイルシステムを定義し，それに対する処理でファイル操作を定義している．FTAM実現に当たっては，この仮想ファイルシステムを実ファイルシステムにマッピングする必要がある．

仮想ファイルシステムのために必要となるすべてのファイル属性は非常に多大である．特にアクセスパスワードやコンテンツタイプなどの属性は，我々が実装したME 1000のOSであるUNIXのファイル管理機能の属性には，対応するものが存在しない．またUNIXではファイル編成という概念がないため，アクセスの対象となるファイル（アクセスファイル）の蓄積情報は，UNIXと別の形で管理する必要がある．こ

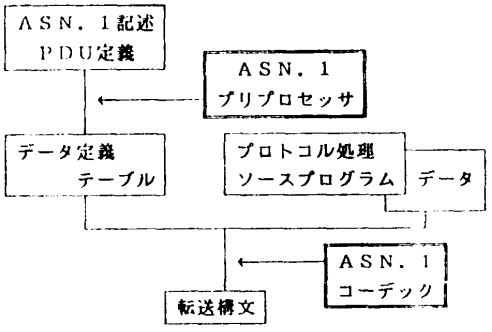

図1 ASN. 1ツールの構成
Fig. 1 Structure of ASN. 1 tools.

のため我々は，仮想ファイルシステムのために必要となるファイル属性情報を，OS のファイル属性情報とは別にファイルとして保存することによって解決した．

別ファイルに保存する方法として，すべてのアクセスファイルの属性を一括して1つのファイルに保存する方法と，アクセスファイルごとに個別のファイルで保存する方法が考えられる．ここでは，ファイルが多数になったときの検索効率を考慮して，後者の方法を採用した．属性保存ファイルの位置としては，アクセスファイルとのアクセス権の独立性から，特定のディレクトリ下に配置した．また，アクセス時の効率を考慮して，ルート以下のディレクトリ構造を再現して，アクセスファイルが位置するディレクトリに対応するディレクトリ下に属性ファイルを配置した．

また，FTAM では構造を持ったファイルは扱えるが，ファイルシステム自体の構造を意識することができないため，UNIX の木構造ファイルシステムを FTAM で処理できない．これを実現するにはディレクトリをファイルとしてアクセスすることにより構造情報を得て，FTAM 利用者が構造を意識する形とした．この点は FTAM の拡張が必要となってくる．

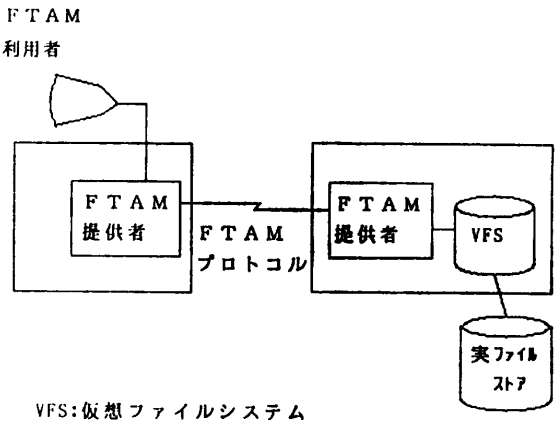

図 2 FTAM のモデル
Fig. 2 Model of FTAM.

4.3 ソフトウェア構成

今回実装した各プロトコルモジュールはすべて superC で記述されているため，オブジェクト指向でいうクラス構成に従ってそのソフトウェアは構築されている．

表 3 に各層モジュールにおけるクラス構成とステップ数を示す．

システム初期化時に，FTAM 管理クラスと FTAM サーバクラスのインスタンスを唯1つ生成する．

表 3 ソフトウェア構成
Table 3 Software structure.

プロトコル	クラス名	ステップ数	機能
FTAM	ユーザクラス	1540	FTAM 利用者にC言語のライブラリインタフェースを提供する
	サーバクラス	990	実ファイルシステムと仮想ファイルシステムのマッピングを行ってファイルにアクセスする
	管理クラス	350	複数のコネクションを管理する
	コネクションクラス	3590	1本のコネクション上の FTAM のプロトコルマシンを実現する状態をインスタンス変数に持ち，入力イベントに応じた処理をメソッドとして持つことで状態遷移を行う
	データユニットクラス	2590	サービスプリミティブおよびプロトコルデータユニットに対する処理を扱う．各データに共通の処理を行うクラスのサブクラスとして各データ固有の処理を行うクラスが存在するという階層構成をとっている
ACSE	管理クラス	220	FTAM と同様
	コネクションクラス	550	FTAM と同様
	データユニットクラス	1190	FTAM と同様
PP	管理クラス	230	FTAM と同様
	コネクションクラス	700	FTAM と同様
	データユニットクラス	1540	FTAM と同様
	符号化／復号化クラス	850	指定された符号化規則に従い，抽象構文と転送構文間の変換を行う

FTAM ユーザクラスは利用者ごとにインスタンスを生成する．FTAM リカバリクラスと FTAM コネクションクラスは，コネクションを確立するごとにインスタンスを生成し，コネクションを解放すると消去する．データユニットクラスは，データを送信する必要があるときに，そのインスタンスを生成する．データの送受信はこのオブジェクトの受け渡しに相当する．他層モジュールにおいても同様である．

このように，オブジェクト指向のクラスとインスタンスの概念を用いることで，ソフトウェア構成をモデルに沿った形で整理でき，容易に実装することができた．

4.4 プロセス構成

プレゼンテーション層および応用層はプロセス間通信によるオーバヘッドを回避するために，1つのプロセスで実現した．また，複数の利用者が存在するホスト計算機のような場合には利用者ごとに各プロトコルモジュールを含んだプロセスが生成されるのは望ましくない．各プロコトルモジュールを複数のコネクションを同時に制御可能とし，FTAM 利用者モジュールと FTAM 以下のプロトコルモジュールは，別プロセスとしても分離可能とした．下位層は，セション層プロトコルモジュールとトランスポート層プロトコルモジュールを1つのプロセスとし，ネットワーク層以下は OS のドライバおよびファームウェアで提供されているものを使用した．

5. 評　　価

ME 1000 を LAN (10 Mbps) で接続して通信を行い，FTAM の転送性能を評価した．LAN 上で評価した理由は，回線上での回線速度の遅さによる影響を受けずに評価するためと，FTP と比較するためである．

評価に先立ち，まずプレゼンテーション層からメモリ上のデータ (1024 bytes) を，セション層のデータ転送要求である S-DATA 要求の利用者データとして

表 4-1 ファイル書き込み時のサービスプリミティブの実行時間
Table 4-1 Execution time for service primitives writing file.

項　目	サービスプリミティブシーケンス	所要時間
FTAM の開始	F_INITIALIZEreq ⟶ F_INITIALIZEcnf ⟵	0.77 sec
ファイルの選択	F_SELECTreq ⟶ F_SELECTcnf ⟵	2.38 sec
ファイルのオープン	F_OPENreq ⟶ F_OPENcnf ⟵	0.63 sec
書き込み要求	F_WRITEreq ⟶	0.08 sec
データ転送 (768 bytes×20)	F_DATAreq ⟶ :	2.07 sec (=7.26 kbytes/sec)
データの終了	F_DATA_ENDreq ⟶	0.12 sec
転送の終了	F_TRANSFER_ENDreq ⟶ F_TRANSFER_ENDcnf ⟵	0.38 sec
ファイルのクローズ	F_CLOSEreq ⟶ F_CLOSEcnf ⟵	0.33 sec
ファイルの解放	F_DESELECTreq ⟶ F_DESELECTcnf ⟵	0.30 sec
FTAM の終了	F_RELEASEreq ⟶ F_RELEASEcnf ⟵	0.42 sec

表 4-2 ファイル読み込み時のサービスプリミティブの実行時間
Table 4-2 Execution time for service primitives reading file.

項　目	サービスプリミティブシーケンス	所要時間
FTAM の開始	F_INITIALIZEreq ⟶ F_INITIALIZEcnf ⟵	0.77 sec
ファイルの選択	F_SELECTreq ⟶ F_SELECTcnf ⟵	3.48 sec
ファイルのオープン	F_OPENreq ⟶ F_OPENcnf ⟵	0.30 sec
読み込み要求 データ転送 (1 kbytes×20) データの終了	F_READreq ⟶ F_DATAreq ⟵ : : F_DATA_ENDreq ⟵	2.32 sec (=6.47 kbytes/sec)
転送の終了	F_TRANSFER_ENDreq ⟶ F_TRANSFER_ENDcnf ⟵	0.28 sec
ファイルのクローズ	F_CLOSEreq ⟶ F_CLOSEcnf ⟵	0.33 sec
ファイルの解放	F_DESELECTreq ⟶ F_DESELECTcnf ⟵	0.32 sec
FTAM の終了	F_RELEASEreq ⟶ F_RELEASEcnf ⟵	0.43 sec

表 5 グルーピングの効果
Table 5 Effectiveness of grouping service primitives.

グルーピングを使用しない場合		
F_CLOSEreq	←——	
F_CLOSEcnf	——→	
F_DESELECTreq	←——	
F_DESELECTcnf	——→	計 0.63 sec
グルーピングを使用した場合		
F_BEGIN_GROUPreq + F_CLOSEreq + F_DESELECTreq + F_END_GROUPreq	——→	
F_BEGIN_GROUPcnf + F_CLOSEcnf + F_DESELECTcnf + F_END_GROUPcnf	←——	
		計 0.58 sec

送信することで，セション層のスループットを測定した．その結果，約 27.3 kbytes/sec であった．

5.1 ファイル転送性能評価

15 kbytes のファイル転送を行い，各サービスプリミティブの経過時間を測定した．書き込み時の結果を**表 4-1** に，読み込み時の結果を**表 4-2** に示す．1回のファイルアクセス単位は 768 bytes であり，結果は10回の測定の平均値である．表 4-1 に示すように，F_SELECT（ファイルの選択）にかなり時間を要している．これは属性ファイルの探索とチェックを行っているためである．また，ファイル転送時の性能は約7 kbytes/sec と判断できる．この結果は 10 Mbps の LAN 上では遅いものとも言えるが，9600 bps 程度の回線上であれば回線速度がネックとなるため十分なものと言える．表 4-2 でもほぼ同様の性能である．

5.2 グルーピングの効果

FTAM では，転送に先立つファイルの選択やオープンのためのサービスプリミティブがすべて確認型である．これらの転送時間を節約するために，複数のデータを1つの PSDU (Presentation Service Data Unit) にマッピングするためのグルーピング機能がある．グルーピングを使用した場合と使用しない場合では，**表 5** に示すように，約 10% 程度の差が生じた．

5.3 superC メッセージ探索の影響

superC では，通常のオブジェクト指向言語で行われている動的束縛（実行時にクラスを探し，メソッドを実行）のほかに，クラス名を陽に指定してメソッド探索に要する時間の短縮を図る静的束縛を用意している．文献[6]によれば静約束縛に要する時間は動的束縛の約2分の1である．静的束縛の効果を測定するために，クラス名の自明な箇所を静的束縛に書き直して，5.1 節と同様の測定を行った．書き換え可能な箇所は全メッセージパシングの約半分であった．

静的束縛を用いたプログラムで表 4-1 と同様の測定を行った結果，約 7.29 kbytes/sec の転送性能であり，特に有意差は生じなかった．これより，プログラムとしての処理の走行時間に比して動的束縛のオーバヘッドはほとんど無視できると言える．すなわち，superC 導入で懸念されたメッセージ探索のオーバヘッドは全体系に影響を与えていないことが判明した．

6. FTP との比較

FTP は米国国防省で ARPANET 用に開発されたプロトコルであり，FTAM と同じく異機種間のファイル転送を行うためのものである．TCP/IP が UNIX 系ワークステーションの標準プロトコルとなったことで，UNIX 系同士あるいは UNIX 系と他 OS 機種とのファイル転送プロトコルとして広く利用されている．

FTAM をファイル転送プロトコルとしての面から評価するために，ME 1000 上に 4.2 BSD 版から移植する形で FTP を実装し，構成，機能および性能からみた比較を行った．結果を**図 3** および**表 6** に示す．これより，以下のことが導かれる．

- FTP の場合は TCP（トランスポート相当）までを OS カーネル内で実現した．これに対し，FTAM の場合は OS で実現されているのはネットワーク層までであり，その上でトランスポートとセションのプロセスと，プレゼンテーション以上のプロセスがプロセス間通信を用いて通信している．この実現形態が性能差

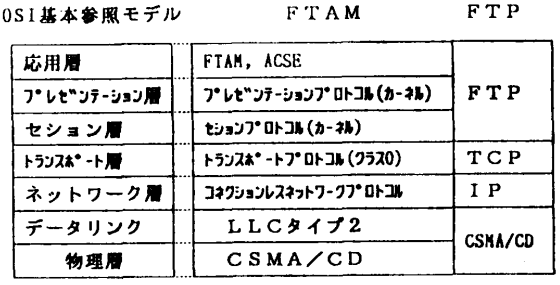

図 3 FTAM と FTP の比較（構成）
Fig. 3 Comparison of FTAM (structure aspect).

表 6 FTAM と FTP の比較（機能・性能）
Table 6 Comparison of FTAM and FTP (function aspect, performance aspect).

側面	項目	FTAM	FTP
機能	ファイルの全体転送	可	可
	ファイル転送の中断・再開	可	不可
	ファイルのアクセス	可	不可
	ファイルの生成／削除	可	可
	構造を持ったファイルの処理	可	不可
	コード変換	有*1	有
性能	秒当りのファイル転送性能	約 7 kbytes	約 30 kbytes

*1 プレゼンテーション層で実現

の大きな原因となっている．

●FTP では一応のコード変換を行ってはいるものの，FTAM ほどの複雑な符号／復号化処理は行っておらず，転送性能にもこの差が多少影響している．

●FTAM では仮想ファイルストアの概念を用いて，構造を持ったファイルを転送したり，ファイルの転送だけでなく，アクセス機能も規定されている．これにより，FTP に比べよりきめ細かい処理が可能である．

7. 考　　察

（1） オブジェクト指向による記述

オブジェクト指向言語 superC の採用により，以下のようなオブジェクト指向の恩恵を受けてプログラムを作成することができた．

●クラスとインスタンスの関係を利用してモデルと整合性のある形での実装が可能であった．そのため，大変読みやすいプログラムとなった．

●データをオブジェクトとしたためにデータ構造が隠蔽され，データに対する操作だけでプロトコル処理モジュールを記述できた．さらに設定するパラメタの追加や変更を外部に影響なくクラスに閉じて行うことができた．

●コネクション管理に関する処理を管理クラスで担当しているためにコネクションクラスの中では規格で規定されているコネクション内のプロトコル処理を忠実に反映させることによって容易に実現できた．

●管理クラスなどの処理は各プロコトルモジュールで共通の処理が多いため，再利用が可能であった．

●データに共通の処理をスーパクラスとして位置づけ，継承機能を利用することができた．

（2） 開発効率

開発の効率に関しては，セションシミュレータが有効であった．マシンの資源や下位層ソフトウェア／ハードウェアの影響を受けること無く，高位層プロトコルのみの試験に集中することができた．さらに superC による記述で，インタフェースを明確に規定することが可能となり，クラスの継承機能や再利用により，COCOMO モデル[8]に比較して約 2 倍の生産性を上げることができた．

（3） 移植性

移植性に関しては superC の移植性の良さにより，アーキテクチャの異なるホストコンピュータ MELCOM-EX シリーズ上への移植においても，コンパイラの相違の吸収，仮想ファイルストアと実ファイルシステムのマッピングを除いて，問題無く実装することができた．

また，プログラムサイズも FTAM 9k, ACSE 2k, PP 3k（ステップ）と小型機にも移植可能な大きさで実現できたため，小型機との OSI 通信も可能という見通しができた．

（4） 性　　能

性能に関しては，複数のプロトコルを 1 プロセスとして実現することにより，約 7 kbytes/sec という値が得られた．これは LAN においては FTP などに劣るが，回線系においては異機種間のファイル転送ということを考慮すれば実用には耐えうる値と思われる．

（5） 拡張性

また，汎用性にも重点を置いて，プロトコルに依存しないクラス構成とし，データ表現も抽象構文の形のままで扱えるツールを開発したために，他応用プロトコル実装時にも同様のアーキテクチャで対処可能と考えている．

（6） 他プロトコルとの比較

さらに FTP との比較では，機能面では細かな制御が可能である反面，数分の一の性能しか得られなかった．これは FTP がトランスポート層相当までカーネルで処理していること，プレゼンテーション層に相当する符号／復号化処理を行っていないことなどが原因と思われる．処理構成や効率的なレイヤ間のデータ転送方法についてさらに検討が必要である．

8. おわりに

本稿では，我々が実装した OSI プロトコルの FTAM, ACSE, プレゼンテーションのソフトウェアについて報告した．次の点については未検討であり，今後

のさらに検討すべき課題である．

●符号化／復号化処理のオーバヘッド：FTAM では転送するファイルデータの構造が比較的単純であったために，符号化／復号化処理が全体系に与える影響は元々小さかったと思われる．しかし，ODIF (Office Document Interchange Format) のように複雑で深い構造を持ったデータを扱う場合には，その符号化／復号化処理が全体の性能にどれだけ影響を与えるか，今回の ASN. 1 ツールで満足な性能が得られるのかなどについて検討する必要がある．

●FTAM フルセットサポートに伴う仮想ファイルストアマッピング部の拡張

●本プログラム構成による他応用プロトコル実装

●7層を通じての総合的なシステム設計：各プロトコル自体は実装できても，それぞれが一貫して動作しなければ全体としてのスループットは向上しない．今回の実装においても，トランスポート層やセション層での分割の単位と応用層からの転送ファイルデータの大きさの相関関係により，処理時間やバッファ使用量の局所化が生じてスループットがかなり変化した．システムの構築に当たり，7層全体を考慮した各プロトコルの選択方法を見いだす必要がある．

謝辞 本論文作成において，九州大学工学部情報工学科牛島和夫教授，荒木啓二郎助教授には助言と示唆をいただいた．ここに感謝する．

参考文献

1) 水野，井手口，市橋，勝山，坂，神原：高位プロトコルを含めたLANシステムの試作，情報処理学会論文誌，Vol. 26, No. 4, pp. 577-583 (1985).
2) ISO: Information Processing Systems-OSI-File Transfer, Access and Management, ISO/DIS 8571 (1987).
3) ISO: Information Processing Systems-OSI-Service Definition/Protocol Specification for Common-Application-Service-Elements-Part 2: Association Control, ISO/DIS 8649-8650/2 (1987).
4) ISO: Information Processing Systems-OSI-Connection Oriented Presentation Service Definition/Protocol Specification, ISO/DIS 8822-8823 (1987).
5) Gref, V. and Cain, E.: The Dod Internet Architecture Model, *Computer Networks*, pp. 307-318 (1983).
6) 勝山，佐藤，中川路，水野：オブジェクト指向型言語 spiceC による通信ソフトウェアの開発，第33回マルチメディア通信と分散処理研究会 (1987).
7) ISO: Information Processing Systems-OSI-Abstract Syntax Notation/Basic Encoding Rule, ISO/DIS 8824-8825 (1987).
8) Boehm, B.: *Software Engineering Economics*, Prentice-Hall, Englewood Cliffs (1981).

(昭和63年2月23日受付)
(昭和63年9月5日採録)

中川路哲男（正会員）

昭和33年生．昭和56年3月東京大学電気工学科卒業．昭和58年3月同大学大学院工学研究科電気工学専攻修士課程修了．同年三菱電機(株)入社．現在同社情報電子研究所システム・ソフトウェア開発部に勤務．OSI通信ソフトウェアを中心とする分散処理システムの構築およびソフトウェア工学に関する研究・開発に従事．電子情報通信学会会員．

勝山光太郎（正会員）

昭和29年生．昭和51年3月大阪大学基礎工学部制御工学科卒業．同年三菱電機(株)入社．現在同社情報電子研究所システム・ソフトウェア開開発部．入社以来通信ソフトウェアの開発，分散処理システムの研究開発に従事．昭和55年よりOSI標準化活動に参加．

芥川 哲雄（正会員）

昭和19年生．昭和42年東京大学電気工学科卒業．同年三菱電機(株)入社．現在同社コンピュータ製作所コンピュータ方式統轄部．コンピュータ・ネットワーク機器のハードウェア，ソフトウェアの方式技術の開発に従事．

水野 忠則（正会員）

昭和20年生．昭和43年名古屋工業大学経営工学科卒業．同年三菱電機(株)入社．現在同社情報電子研究所システム・ソフトウェア開発部．工学博士．情報通信システムおよび分散処理システムに関する研究・開発に従事．著書としては『マイコンローカルネットワーク』(産報出版)，『分散処理システム・デザイン』(共訳，工学社)，『電子メールとメッセージ通信』(監訳，工学社) などがある．電子情報通信学会，オフィスオートメーション学会，日本経営工学会，IEEE 各会員．