

OS インタフェースのサブセット化

柴垣 斉 和佐野 哲男

日本電信電話(株) ネットワークシステム開発センタ

種々の適用分野の要求に応じた必要最小限のインタフェースからなるOSを構築するためには、OS全体のインタフェースから必要なインタフェース部分を切り出すサブセット化が必要である。本論文では、アプリケーションプログラムのポータビリティとOSインタフェースの多様な選択可能性を両立させるサブセット化手法として以下を提案する。①OSインタフェースをOSが制御対象とする資源に着目して、複数の独立した単位に分類する。(これをインタフェース単位と呼ぶ。)②インタフェース単位毎にサブセットを規定する。③適用分野毎のサブセットの組み合わせをプロファイルとして規定する。

OS INTERFACE SUBSETTING

Hitoshi SHIBAGAKI, Tetsuo WASANO

NTT Network Systems Development Center

1-2-1, Uchisaiwai-cho, Chiyoda-ku,

Tokyo, 100, Japan

In order to allow each OS to be implemented with the minimum functions in the particular application field, it is necessary to divide the OS into interface subsets. This paper reports on the three-stage subsetting method described below, which is aimed at achieving AP portability while also providing a wide choice of OS interfaces.

- (1) OS interfaces are classified into a number of independent units based on the resources that are controlled by the OS. These are called interface units.
- (2) In each interface unit, interface subsets are specified.
- (3) Combinations of subsets are specified for each application field as profiles.

1. はじめに

現存する汎用オペレーティングシステム（以下OS）は、その開発当初は限定されたアプリケーションプログラム（以下AP）用に開発されたが、時間の経過とともに多種多様なAPの動作を可能とすべく拡張されてきた。この結果、OSとして提供するインタフェースは豊富となったが、個々のAPからみると不要なインタフェースも多く、性能、使いやすさの点で問題が生じてきた。これに対処するためには、OSインタフェースを分類、体系化したうえで、適用先APが必要とするインタフェースのみの選択を可能とするOSインタフェースのサブセット化が必要である。

一方、ソフトウェア需要の増大、ソフトウェアの複雑化、高価格化からソフトウェアの再利用、すなわちポータビリティの確保はますます重要となっている。この観点からOSインタフェースのサブセット化を見た場合、システムによって選択したOSインタフェース（APの動作環境）が異なると、APは他システムでの動作が不可能となる、すなわちポータビリティが損なわれるという問題が生じる。このため、OSインタフェースの選択には、ポータビリティを損なわないよう一定の基準が必要である。

異なったシステム間の相互接続性、協調動作を可能とするための必要な機能の組み合わせはISP (International Standardized Profiles) ⁽¹⁾ としてその概念が検討されている。OS Iシステムの相互接続性を確保するため通信プロトコルのプロファイルへの適用として成果がでているが、OSインタフェースへの適用はまだ行われていない。またOSのサブセット化については一部検討されているが⁽²⁾、カーネルインタフェースの範囲にとどまり、OSインタフェース全体への適用はなされていない。

本論文では、APのポータビリティとOSインタフェースの多様な選択可能性を両立させるサブセット化手法について報告する。なお本論文で述べるサブセット化手法はCTRON ^{(3) (4) (5) (6)}

の設計において採用され、サブセット化仕様の決定に適用された。

2. サブセット化手法

本論でのサブセット化とは、各分野の要求に応じて、必要最小限のインタフェースからなる目的対応のOSを構築するため、OS全体として規定するインタフェースから必要なインタフェース部分を切り出すことである。ポータビリティを損なわない範囲で必要最小限のインタフェースを切り出すため以下の三段階のサブセット化手法を提案する。

- ① OSインタフェースをOSが制御対象とする資源に着目して、複数の独立した単位に分類する。（これをインタフェース単位と呼ぶ。）
- ② インタフェース単位内で、インタフェースプリミティブ（システムコール）を機能的なまとまりの組（サブセット単位）に分類し、さらにOSをインプリメントする際に必要なサブセット単位の組み合わせをサブセットとして規定する。
- ③ 各インタフェース単位毎に定めたサブセットをOSインタフェース全体として体系化し、適用分野毎のサブセットの組み合わせをプロファイルとして規定する。

概要を図1に示す。以下、この手法の詳細を説明する。

3. インタフェース単位

OSの役割は、コンピュータを構成する物理資源（ハードウェア資源）、論理資源（ソフトウェア資源、データベース資源）を管理し、

- ① ユーザが資源を上手に（使用効率を良く、使い易く）利用できるように、多様な資源毎の特性を仮想化する機能とそれらの資源へのアクセス手段、
- ② 複数ユーザによる資源の共用（資源の配分、セ

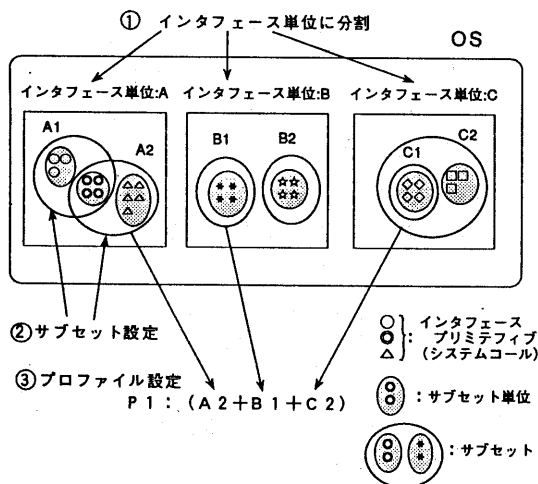


図1 サブセットとプロファイル

セキュリティ)を可能とする機能

さらに、広義には、

③ ユーザの問題解決を行うための共通的な機能
(数値処理のライブラリ関数、各種ユーティリティなど)

に大別できる。

物理資源の例は、中央処理装置、主記憶装置、通信制御装置、端末、プリンタなどの入出力装置、磁気ディスクや磁気テープなどの2次記憶装置などである。論理資源の例は、プログラムやデータ、ユーザから見た仕事の単位であるジョブやセッション、中央処理装置(処理の時間)、主記憶装置(記憶アドレス)などの資源の割当て対象となる

コンピュータ内での仕事の単位であるタスク、さらにプログラムやデータの2次記憶装置内での格納域であるファイルなどである。これらの資源種別に着目し、OSインタフェースを分類・整理することを試みた。

以下、CTRONで採用した具体例を示す。

① OSインタフェースの分類

OSが管理対象とする資源の明確化を行い、分類する。そして、その分類に対応するOSインタフェース群をインタフェース単位と定義する(表1参照)。OSやAP設計者は、まずは対象とする資源の要否に基づき所要インタフェースの範囲を特定できる。

② 二階層構成

OSインタフェースを、物理的な資源の仮想化を行うインタフェース階層と論理的な資源に対応するインタフェース階層よりなる2階層構成を持つものとする。前者は、ソフトウェアのポーティングを行うためのプラットフォームとして最も基本的なOSインタフェースを提供する。このインタフェース階層を実現するOSソフトウェアは、物理的な資源毎の様々なアーキテクチャを意識した制御を行うことからアーキテクチャ種別毎にインプリメントすることになる。例えば、中央処理装置といっても機種が違えば、機種毎に独自のソフトウェアとしてインプリメントしなければならない。このインタフェースの上位に、ファイル、プロセス、ジョブなどの、より論理的な資源に対応するインタフェース階層を位置づける。このイ

本論文で用いる主な用語の定義を以下に示す。

インタフェース : OS機能を使用するためにOSがAPにみせる切り口。

インタフェース単位 : OSインタフェースをOSが制御対象とする資源に着目して分類した単位。

インタフェースプリミティブ : インタフェースの最小単位。システムコールが相当する。

サブセット単位 : インタフェースプリミティブを機能的なまとまりに分類した単位。

サブセット : サブセット単位の集合でインプリメントの単位となるもの。

プロファイル : サブセットの組み合わせでOSシステム全体を構成するもの。

表1 CTRONでのインタフェース単位と管理資源

分類	インタフェース単位	管理資源	
基本インタフェース階層	カーネル	プロセッサ、メモリ	
	入出力制御	入出力デバイス	
上位 インタフェース 階層	情報蓄積制御	ファイル管理	ファイル
		データベース管理	データベース
	通信制御	通信制御共通部	通信用バッファ、タイ
		ネットワーク層制御	ネットワーク層通信パス
		トランスポート層制御	トランスポート層通信パス
		セッション層制御	セッション層通信パス
		以下略	
	交換制御	ISDNユーザ制御	交換用通信パス
	実行制御	プログラム管理	プログラム
		プロセス管理	プロセス
	保守運転管理	保守運転管理	管理資源

インタフェース階層のソフトウェアは、異なる機種上にインプリメントされた基本インタフェース階層OS間でのポーティングが実現可能となる。

4. サブセット

サブセットは、それぞれのインタフェース単位の中で、目的とする適用分野において要求されるインタフェースプリミティブの集合である。有効とするサブセット(*)の種類を増やすと、その数だけ提供インタフェースの異なるOSがインプリメントできることになり、個々のAPは自分に最適なOSを利用することができる。一方、特定のOSのみが持つインタフェースを使用するAPは、他のOS上では動作しない。つまり、サブセット条件の異なるOS間ではソフトウェアのポータビリティが失われる。ポータビリティは、ソフトウェアの再利用や共用によりソフトウェア生産性の

向上を可能とするための重要な要素である。このように、サブセットの種類が増大は、個々のAPへのチューニングというメリットをもたらす反面、OS間でのソフトウェアのポータビリティを阻害するデメリットを招くことになり、このサブセット規定の設計ではAPへのチューニングとソフトウェア生産性のトレードオフ問題を解くことになる。

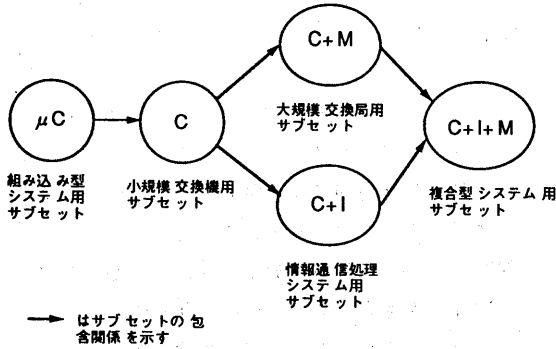
まずインタフェース単位内のインタフェースプリミティブを機能的なまとまりごといくつかの組(サブセット単位)にわけ、次に適用分野を考慮していくつかのサブセット単位を組み合わせるサブセットを設定する。

サブセットの設定にさいしては以下の原則を適用する。

- ①すべてのインタフェースプリミティブはいずれかのサブセット単位に属し、また所属するサブセット単位はただ一つとし、重複は許さない。
- ②サブセット単位の規模は、さらに細分化する要求が見えない範囲で最大とする。これは組み合わせの種類の数に不要な増大を防ぐためである。
- ③サブセットの種類を増やすことはポータビリティの阻害要因である。従ってその数はAPへの適用性が損なわれない範囲内で必要最小限とする。
- ④あるサブセットのインタフェースを使用したAPが、他のサブセット上で動作可能か否かを判断できるようにするため、サブセット相互間の包含関係を明確にする。
- ⑤現在運用に供されているシステムのOSインタフェース、及びその将来計画を調査し、必要インタフェースを抽出したのち上記原則に基づきサブセットを設定する。この際、OSインプリメンタ側の知識のみでなく、ユーザの要求をできる限り集めたうえで決定する。

CTRONで採用したサブセット例を図2に示す。

(*) 本論において「サブセット」の用語は、集合論でいう部分集合の全体ではなく、有効な部分集合のみをさすものとして用いている。



サブセット単位：記号	機能	インタフェースプリミティブ数
μC	タスク管理 タスク同期通信(イベントフラグ、メッセージボックス) 割り込み管理 時計管理 メモリ管理	44
C-μC	タスク同期通信(セマフォ、シリアルリユーザブル資源) 例外管理 統計情報管理	40
M	高速処理用タスク制御 タスク同期起動制御 メッセージ選択受信制御 ランデブ制御 私用タイマ制御	19
I	仮想記憶制御 ロールイン/ロールアウト制御 基本プログラム管理	22

図2 サブセット (CTRONカーネルの例)

5. プロファイル

5.1 目的と設定方針

実際のシステムでは、OSはインタフェース単位毎のサブセットをいくつか組み合わせた形で提供される。APは一般に複数のインタフェース単位を使用するため、APのポータビリティを高めるためには、提供すべきサブセットの組み合わせ、すなわちプロファイルを決める必要がある。

プロファイルの設定にあたって以下の原則を適用する。

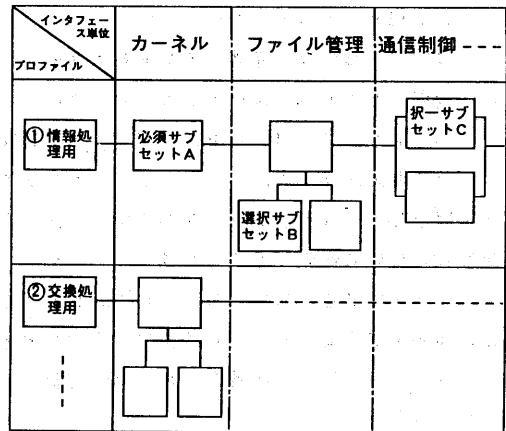
① 適用分野に応じた最適な組み合わせとするため、プロファイルも適用分野毎に定める。ただし、異なるプロファイル間ではポータビリティに制約が生じることから、プロファイルの種類は必要最小限とする。

② あるインタフェース単位を使用する際、どのサブセットを使用するかをプロファイル規定として定める。この際、それぞれのサブセットを、必須(提供が必須なもの)、択一(いずれか一つの提供が必要なもの)、選択(提供するか否か自由なもの)に分類して規定する。

③ あるサブセットを提供するために必須となる他のインタフェース単位のサブセットが存在する。これらサブセット間の関係も考慮して設定する。

④ ポータビリティ向上のためには必須サブセットをなるべく多く、逆にコンパクトなOSのインプリメントのためにはなるべく少なくする必要がある。4章と同様、ユーザ、OSインプリメンタ両者の要求を調査し設定する。

プロファイルの記述法を図3に示す。



必須サブセット：規定するプロファイルにおいて提供が必須なサブセット
 択一サブセット：いずれか一つの提供が必要なサブセット
 選択サブセット：提供するか否か自由なサブセット

図3 プロファイルの記述法

5. 2 プロファイルの設定

前節で述べたように、プロファイルは適用分野毎に設定する。適用分野はAPのポーティング範囲がなるべく広がるように、APの共用が想定される範囲で設定する。

以下の手順でプロファイルを定める。

①インタフェース単位毎に各対象分野に最適なそれぞれのサブセットを抽出する。

②サブセットの組をプロファイルとしてまとめ、サブセット間の関係(サブセット相互で提供を前提とするものが含まれているか)に問題がないかチェックする。

③作成したプロファイルをOSのインプリメンタ、AP作成者、システム設計者のそれぞれの立場から、インタフェースの過不足をチェックする。

④指摘された問題点をプロファイル規定に反映する。

プロファイルの詳細例を図4に示す。

5. 3 プロファイル設定上の主要な考慮点

(1) 選択サブセットの扱い

プロファイルには、提供するか否かを自由とした「選択」サブセットが存在する。同一プロファイルのシステムでも同一の選択サブセットを提供している保証はないため、ポーティングを考えるAPはなるべく選択サブセットのインタフェースを使用しないで作成することが望ましい。選択サブセットを使用せざるをえない場合は、使用するAP部分を切り離してOSの選択サブセットとともにポーティングさせる構成とするのが望ましい。

(図5参照)

ポータビリティを考慮するうえでは必須サブセット、択一サブセットが最も意味をもつ。

(2) プロファイル間のポーティング系列

あるプロファイル上で動作するAPが他のプロファイル上で動作するか否かが明確になるようにプロファイル間のポーティング系列を定める。但し、このポーティング系列は必須サブセットのみ

を使用する場合に有効である。例を図6に示す。矢印の方向にAPのポーティングが可能となる。

(3) AP、OSインプリメント上の考慮点

ポーティングを考えるAP作成者は、そのAPが動作する適用範囲を考慮し、その分野でのプロファイルのインタフェースのみを使用することが望ましい。また複数の適用分野にまたがったのポーティングを考える場合には、複数のプロファイルに共通に含まれる(AND条件)サブセットのみを使用すれば可能となる。

一方、OSインプリメンタはその適用範囲を考慮し、プロファイルに従ったOSインタフェースの提供が望ましい。複数の提供分野を考える場合は、両者のプロファイル(OR条件)を提供する必要がある。また自己の性能条件等を考慮したうえでどこまで選択サブセットを提供するか判断することとなる。

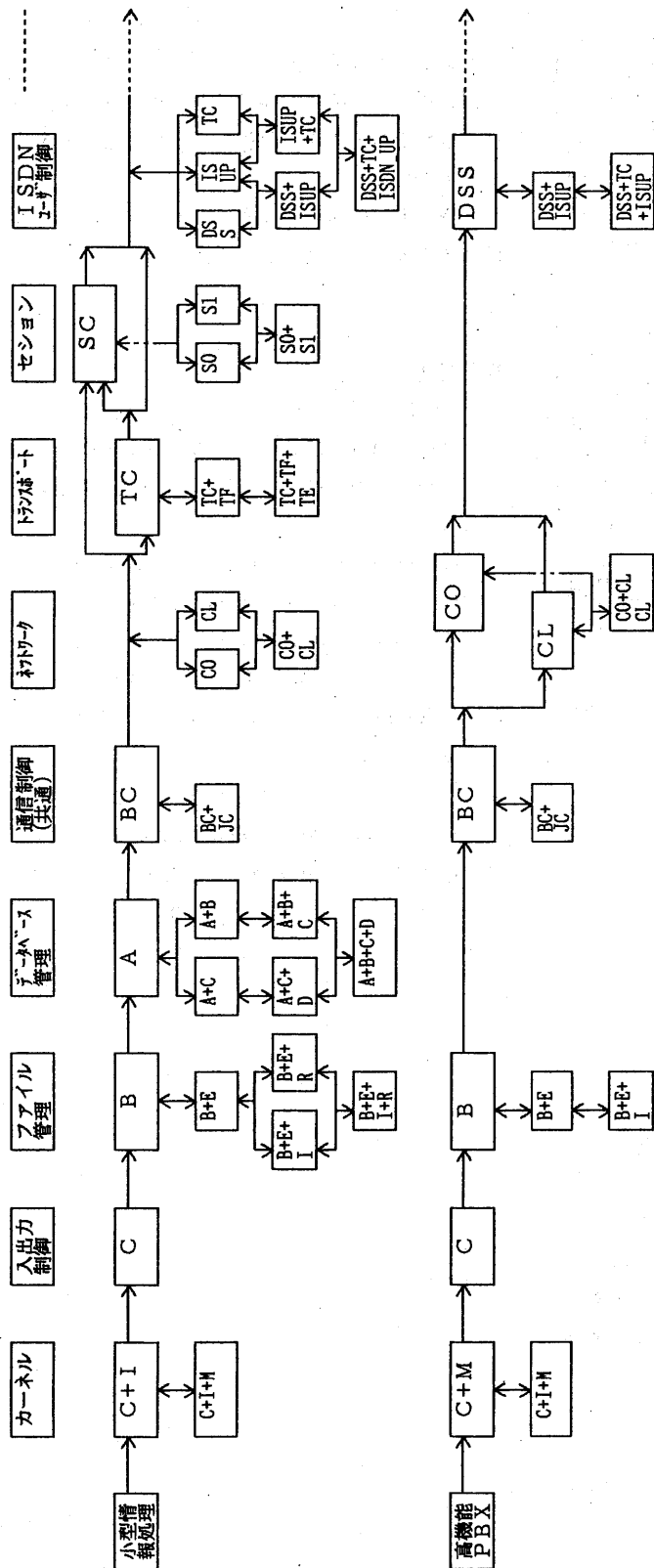
(4) コンフォーマンス

APのポーティングを確実なものとするためには、インタフェース単位毎のコンフォーマンス検定(サブセットを対象とし、インタフェースの過不足、修正の有無をチェックする)に加えて、プロファイルとしての検定を行うことが望ましい。サブセット検定がソフトウェア製品単位に行われるのに対し、プロファイル検定はOSソフトウェアを組み合わせたシステム単位に行われることになる。

6. おわりに

OSインタフェースのサブセット化の必要性、手法についてのべた。

本論文でのべたサブセット、プロファイルの有効性については、実際にどの程度ソフトウェアがポーティングされたかによって評価する必要がある。この際、①同一プロファイル間でのポータビリティ(プロファイルが全く同じでも解釈の違い等によりポーティングが困難となることがある)、



記号一覧

記号	説明	カーネル	入出力制御	ファイル管理	データベース管理	通信制御(共通)	ネットワーク	トランスポート	セッション	ISDNユーザ制御
1	記号説明	C+I マイクロ 機	C 共通部	B 基本編成基本部	A 共通部	BC バツプア管理	CO コネクション型	TC 共通部 (07X0)	SC 共通部	DSS ダイヤル加入者番号
2	記号説明	C 標準		B+E 基本編成拡張部	A+B DML拡張機能	JC 時計管理	CL ダイヤル型	TC+TF 優先ダイヤル(07X2)	SO SC+ダイヤル化ダイヤル	ISDN UP ISDNダイヤル
3	記号説明	C+I 高機能		B+E+I 索引編成ダイヤル	A+C 水準1DDL	BC+JC フルセット	CO+CL フルセット	TC+TF+TE ダイヤル制御 (07X4)	S1 SC+同期系他	ISDNダイヤル ダイヤル制御機能
4	記号説明	C+M 仮想記憶		B+E+R ダイヤル編成ダイヤル	A+C+D DDL拡張機能				SO+S1 フルセット	DSS+ISDN UP 上記組み合わせ
5	記号説明	C+I+M フルセット		B+E+I+R フルセット	A+B+C 上記の組み合わせ				フルセット	TC+ISDN UP 上記組み合わせ
6	記号説明				A+B+C+D フルセット					DSS+TC+ISDN UP フルセット

図4 プロファイルの詳細(例)

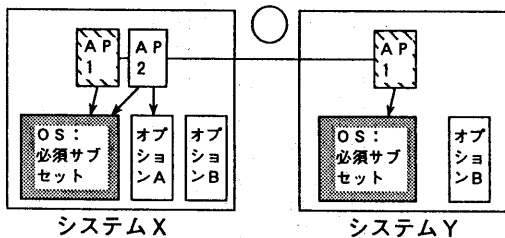
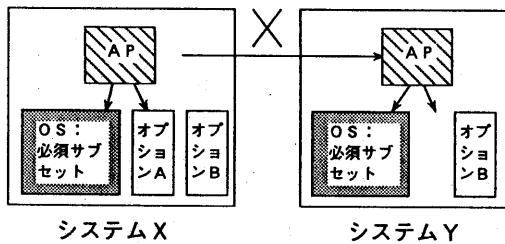
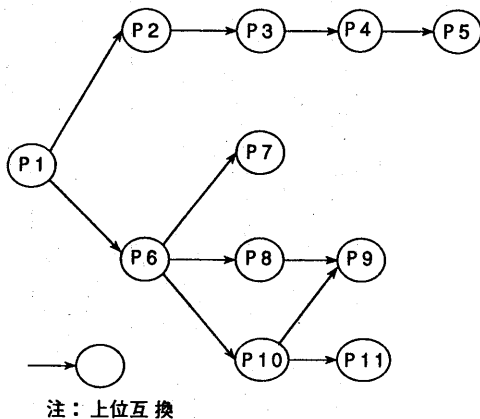


図5 ポーティングを考慮したAPの構成



注：上位互換

図6 プロファイル間の関係

②同一プロファイル内ポーティングにたいする選択サブセットの影響、③異なるプロファイル間でのポーティング、の3点の評価を行う必要がある。

本来サブセットやプロファイルはOSの利用者であるユーザの要求によって決めるべきものであり、また計算機技術の発展により必要なOS機能、インタフェースも変化するため、プロファイルも固定的なものでなく改善していくべきものである。

しかしながらこういったサブセットやプロファイルを構成するうえで必要なOSインタフェースの分類、体系化、そしてサブセット化手法そのものは今後も活用できる一般的手法として確立できたと考えている。

今後、上記問題について、実際の適用の中での評価をふまえて検討を進める。

参考文献

- (1)ISO/IEC：“Information Technology - Framework and Taxonomy of International Standardized Profiles”,ISO/IEC TR 10000-1, 1990
- (2)IEEE Computer Society:“Draft Standard for Microprocessor Operating System Interfaces”, P855/Draft 7.1, May 1988
- (3)トロン協会編:「CTRON概説」、オーム社、1988
- (4)トロン協会編:「カーネルインタフェース」、オーム社、1988
- (5)I.Wasano, et al.:“CTRON Reference Model”, Proc. of 5th TRON Project Symposium, 1988
- (6)和佐野哲男、小林吉純:「情報通信ネットワーク向きOSインタフェースCTRON」、情報処理, Vol.30, No.5, 1989