

マイクロ・メインフレーム結合における仮想サービス

平賀瑠美

日本アイ・ビー・エム株式会社
東京基礎研究所

マイクロ・メインフレーム結合により、パーソナル・コンピュータのユーザが、利用可能な資源やサービスをホスト・コンピュータ上にまで拡張することができるようになった。マイクロ・メインフレーム結合のサービスでは、パーソナル・コンピュータのユーザが、異和感のないインタフェースでホスト・コンピュータをアクセスできるような機構が必要となる。ここでは、ホスト・コンピュータとパーソナル・コンピュータとを結合する際に解決しなければならない問題を、我々が開発した日本語 PC/VM BOND を例にとり述べる。また、より汎用的な機能を提供する ECF を紹介し、マイクロ・メインフレーム結合のあり方についても考える。

“The Virtual Services in Micro-Mainframe Link Software” (in Japanese)

Rumi Hiraga

Tokyo Research Laboratory
IBM Japan, Ltd.
5-19, Sanbancho, Chiyoda-ku, Tokyo 102, Japan

Micro-Mainframe Link enables PC users to utilize large amount of resources and services of host computer. It must give PC users the same user interface as the DOS to access host computer. In this paper we describe the functions of Japanese PC/VM Bond and ECF, and some problems solved in creating Micro-Mainframe Link software by making reference to Japanese PC/VM Bond we developed.

I. はじめに

ホスト（ホスト・コンピュータ）とパソコン（パーソナル・コンピュータ）とを結びつけようとする MML¹のソフトウェアには、多くのものが現れるようになった [萩野85] [林86-1] [林86-2] [石田86] [関山86]。しかし、MMLのソフトウェアは、単に、ホスト-パソコン間でファイルを変換、転送するという機能に限定されているものが多い。MMLは、ファイル転送機能のみに限らず、ホストの資源やサービス、パソコンのユーザ・フレンドリなインタフェースといったそれぞれの特性を生かすために、さらに発展した形態となるべきものである [真田86]。

本稿では、発展した形の MML ソフトウェアである日本語 PC/VM BONDと ECF²が提供するサービスについて述べる。また、それらのサービスをパソコンとホストという異なる環境上に実現する際に生ずる問題とそれに対する解決方法を日本語 PC/VM BONDを例にとり述べる。

2. マイクロ・メインフレーム結合とは何か

ここでは、MMLの出現が要求された背景と、ユーザ・インタフェースから見た MMLの実現方法について述べる [Auerbach85] [Goldstein84] [Parr85]。また、発展した形態の MML ソフトウェアの例として、日本語 PC/VM BONDと ECFの概要を紹介する。

ホストとパソコン

ホストとパソコンは、今までそれぞれの機能的役割を明確に与えられ、オフィスにはほとんど独立に導入されてきた。MMLの出現が要求された背景を考えるにあたり、まずホストとパソコンの役割を分類してみる。

ホスト

- 大きなハードウェア資源を持つ。
 - 高速処理能力
 - 大容量で高速なディスク装置
 - 高速で高機能なプリンタ
- 豊富なソフトウェア資源を持つ。
 - 今までのアプリケーション・プログラムの蓄積
 - データベース
- 共用資源の集中管理に向いている。
- セキュリティ機能が高い。
- 大規模なネットワークに接続されている。
- バッチ処理向きである。

パソコン

- 応答性の良いユーザ・フレンドリなインタフェースを実現できる。
- それほど大きくはないが、個人専用のハードウェア/ソフトウェア資源を持つ。
- 理解し易さ、使い易さを基本に設計されている。
- システム構成の変更が容易である。

この両者の比較は必ずしも厳密なものではない³。ホストとパソコン両者の特長を生かすことにより、オフィスに既存のコンピュータの機能を、新たなハードウェアの導入なしに拡張しようというのが、MMLである。つまり、MMLは、可用性があり使い勝手の良いパソコンから、高機能、大容量の資源を持つホストを使用するという形態を作り出すことで、オフィスのコンピュータの機能を拡張している。

MMLの実現方法

一般にホストとパソコンとは、システム環境（アーキテクチャ、オペレーティング・システム、文字コードなど）が異なっている。したがって MMLが、ホストとパソコンとを結合するためには、二つの異なるシステム環境の違いを解決する必要がある。ユーザに最も身近な環境として、ユーザ・インタフェースを考えると、MMLの実現方法として、次の三つのアプローチがあるように思われる。

1. パソコンとホスト両方の既存のインタフェースをそのまま提供する。ユーザはパソコンの

1 Micro-Mainframe Link=マイクロ・メインフレーム結合。本稿では、「MML」と略す。

2 IBM S/370 to IBM Personal Computer Enhanced Connectivity Facilities = システム/370 - ワークステーション 拡張接続機構。本稿では、「ECF」と略す。

3 例えば LAN（ローカル・エリア・ネットワーク）はパソコン同士を接続してホスト的な機能（データや高価なハードウェア資源の共有）を特定のパソコン（サーバ）に持たせることが多い。

モードとホスト端末エミュレータのモードを切り換えながら使い、ファイル転送によってデータの交換を行なう。

2. 新しい統合 OA 環境用のインタフェースを提供する。
3. パソコンの既存のインタフェースのみ提供する。

アプローチ1では、ユーザが、ホストとパソコンを使うためには、両方のインタフェースを習得しなくてはならない。このようなアプローチは、MMLとしては、初期的なもので、本当にホストとパソコンの特長を生かしているとは言い難い。アプローチ2と3は、ユーザに単一のシステム・イメージを与えることができ、さらに先に述べたホスト、パソコン両者の特長を生かすこともできる。アプローチ2は、統合OA環境のアプリケーションに対して、最適化されたMMLを実現することができるが、アプローチ3のように、既存のパソコンのアプリケーションからホストの資源にアクセスすることはできない。

日本語 PC/VM BOND (および ECF) は、アプローチ3によって MMLを実現しており、DOSのユーザやアプリケーション・プログラムは、ホストの存在を意識することなく、ホストの資源やデータをあたかもパソコンのものであるかのように使うことができる。例えば、ECFの仮想ファイル機能によって、ユーザはパソコンの既存のエディタからホストのファイルを直接編集することが可能である。発展した形態のMMLとは、このようにパソコンのユーザ・インタフェースのみで、ホストの資源やサービスにアクセスできるようになったものを指す。

日本語 PC/VM BONDと ECF

日本語 PC/VM BONDは、パソコンとしてマルチステーション 5550 ファミリー⁴の日本語 DOS⁵と、ホストとして IBM システム/370のVM/SP-CMS⁶との間のMMLを実現するソフトウェアである。3270 端末エミュレータには、3270 PC⁷を使う。

4 IBM 5540、5550、および5560のことをいう。本稿では「5550」で代表させる。

5 日本語 Disk Operating System. 以後、本稿では「DOS」と呼ぶ。

6 Virtual Machine / System Product - Conversational Monitor System. IBM システム /370 アーキテクチャ・マシンのオペレーティング・システムの一つ。本稿では「VMS」と略す。

7 日本語 3270 パーソナル・コンピュータ および日本語 3270 パーソナル・コンピュータ/グラフィックスを指す。日本語 DOS のセッションと2つまでの3270 端末セッションを提供するソフトウェア=エミュレータである。ユーザは、特殊なキー操作でセッションを切り換えることができる。本稿では総称して「3270PC」と呼ぶ。

8 Multiple Virtual Storage/Extended Architecture - Time Sharing Option/Extensions

日本語 PC/VM BONDは、IBM トーマス・J・ワトソン研究所の研究成果を製品化した形で1984年に発表されたIBMパーソナル・コンピュータ用のPC/VM BONDがもとになっている。米国のPC/VM BONDに対して、筆者のグループは日本語処理機能の追加と端末エミュレータの変更を加えて日本語PC/VM BONDの作成を行なった[相原87][平賀87][黒沢87-1][黒沢87-2][KURO87][BOND86]。

PC/VM BOND (日本語 PC/VM BOND) は、ECFの前段階の製品として位置付けられている。ECFでは、パソコンとホストとを結合する仮想サービスをより一層強化し、ユーザがMMLのソフトウェアを簡単に記述できるようにインタフェース=SRPI (Server-Requester Programming Interface)を提供している。

ECFは、ホストのオペレーティング・システムとしてVMのほかに、MVS/XA-TSO/E⁸の下でも使用できる。ECFを稼動させるためのホストのオペレーティング・システムや端末エミュレータは、ECFサポートの機能としてSRPIを含んでいる。

3. 仮想サービス

仮想サービスは、ホストの資源(ディスクやプリンタなど)をあたかもパソコンのものであるかのようにパソコンのユーザに見せたり、アクセスしたりすることを可能にする機能である。MMLソフトウェアである日本語 PC/VM BONDと ECF が提供するサービスについて述べる。

日本語 PC/VM BONDの機能

日本語 PC/VM BONDのサービスには、次のものがある。

- 仮想ディスク機能
- ホスト・コマンド実行機能
- メッセージ交換機能

仮想ディスク機能

仮想ディスク機能は、VMのディスク装置上に仮想的なDOSのディスクドライブを作り、DOS側から実際にパソコンの持っているディスクドライブと同じ方法でアクセスできるという機能である。仮想ディスクの実体は、DOSのディスクットのフォーマットで書かれたCMSファイルである。したがって、仮想ディスクを他のユーザーと共有して⁹使うことができる。また、仮想ディスクはVMによって管理されているため、セキュリティやメンテナンスの点で優れている。

ユーザーは、必要に応じて2キロバイトから31メガバイトまでの大きさの仮想ディスクを作ることができる。仮想ディスクをドライブ指定子に割り当てれば、DOSからアクセスすることができるようになる。ユーザーは、同時に8個まで仮想ディスクを使うことができる。また、仮想ディスク上のDOSファイルとCMSファイルの文字コードとフォーマットを変換して¹⁰、既存のCMSファイルをDOSファイルとして使用することなどもできる。

ホスト・コマンド実行機能

日本語PC/VM BONDのユーザーはDOSコマンド¹¹を使って、DOSの環境からホストのコマンド¹²を実行し、結果をDOSの画面に表示することができる。

メッセージ交換機能

日本語PC/VM BONDのユーザーは、漢字など2バイト・コードを含む日本語のメッセージをDOS側から、VMを通して交換できる¹³。ユーザーは、到着したメッセージを都合のよいときに、簡単なキー操作¹⁴でウィンドウに表示して読み取ることができる。メッセージを読み取ったあとは、同じキー操作をもう一度行なうことで、ウィンドウは消えてもとの画面が回復される。メッセージを表示するため

に特別のプログラムを起動する必要はない。したがって、ユーザーは、DOS側での仕事を中断することなく、メッセージを受け取ることができる。

ECFの機能

ECFは、日本語PC/VM BONDが持つ機能の他に下記のようないくつかの仮想サービスがある。

- 仮想ファイル機能
- 仮想プリント機能
- ホスト・データ・アクセス機能

また、SRPIにより、ユーザーが分散処理プログラムを容易に作ることでできるインタフェースを提供している。

仮想ファイル機能

仮想ファイル機能は、ホストのファイルをDOSのファイルのように見せる機能である。この機能により、DOSファイルのように見えたホストファイルに対してDOSコマンドを実行することができる。

仮想プリント機能

仮想プリント機能を用いれば、パソコンから直接ホストのプリンタ (IBM 1403 や IBM 3800) に出力できる。

ホスト・データ・アクセス機能

ECFのホスト・データ・アクセス機能によって、パソコンから直接ホストのデータを抽出することが出来る。パソコンのユーザーは、ホストのファイルの編成を知る必要はない。この機能では、VSAM (Virtual Storage Access Method) や SAM (Storage Access Method) でアクセスするデー

9 VMのミニディスクのLINK機能を用いる。

10 PCDISKユーティリティ。このファイル変換機能は、3270PCのファイル転送ユーティリティと同等の機能を提供する。

11 VMという名前のDOSコマンド。

12 CMSコマンド、CPコマンド、またはEXECプロシージャ。

13 TELLというDOSコマンドでメッセージを転送する。このDOSコマンドは、VMのメッセージ転送コマンド (TELLコマンド) を起動する。

14 前面キー・Ctrlキー・左シフトキーの3つを同時に押す。

タや、DB2(DATABASE2), SQL/DS(Structured Query Language/Data System)上のデータをアクセスすることができる。

SRPI

SRPIは、リクエストとサーバという分散処理のアプリケーション間のインタフェースである(図1)。リクエストは、パソコン上のアプリケーション・プログラムで、ホスト上のサーバに対し、サービスやデータを要求し、サーバがそれに応じる。ユーザは、SRPIを使用することで、パソコン-ホスト間の通信や、サーバがどこにあるのかを意識せずに、リクエストやサーバを作ることができる。リクエストが、要求するサービスの名前を知っているだけで、パソコンとホストのルート機能により、自動的にその要求に応じたサーバから結果が得られるのである。

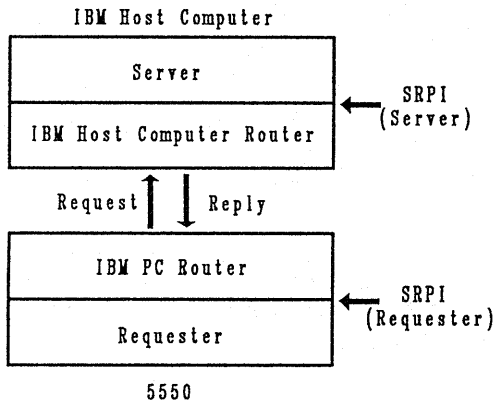


図 1: ECF の論理構造

4. 日本語 PC/VM BONDの基本構成

日本語 PC/VM BONDを一つの MML ソフトウェアの実現例として、その基本構成と文字コード変換について述べる。

システム構成

図 2に日本語 PC/VM BONDの論理構成を示す。パソコン側には VM BOND プログラムがある。これは、リクエストの役割をし、VM に接続している。ホスト側には PC BOND プログラムがあり、サーバとして機能している。基本的な動作は、VM BOND プログラムが出した要求に対して、PC BOND プログラムがその要求を処理して結果をVM BOND プログラムに返すという形をとる。

VM BOND プログラムの実体は DOS の装置駆動ルーチンの集まりであり、PC BOND プログラムの実体は CMS の中核拡張域に常駐するコマンド・モジュールである。VM BOND プログラムは3270PCの API¹⁵によって、PC BOND プログラムとデータの送受信を行なう。また、VM 側にはPCDISKユーティリティがあり、先に述べたように仮想ディスク上のDOSファイルとCMSファイルとの変換などを行なう。

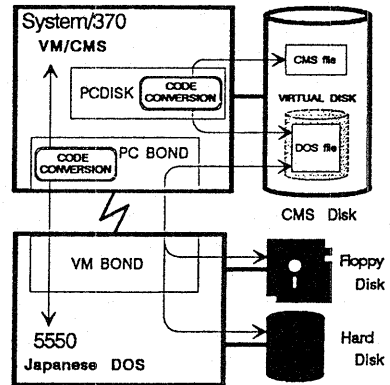


図 2: 日本語 PC/VM BONDの論理構成

コード変換方式

日本語 PC/VM BONDにおけるコード変換について述べる。表 1 に示されるように、DOS と VM とでは異なる文字コード・セットを用いており、ホストとパソコンを連携させるためには、文字コードの変換が必要となる¹⁶。

15 Application Program Interface. (適用業務プログラム・インタフェース)

表1. パソコンとホストにおける文字コード・セットの違い

	1バイト・コード	2バイト・コード
3270PCのPCモード (日本語DOS)	JIS 8ビット・コード	JISCII漢字コード
ホスト・システム (VM/CMS)	EBCDIC 英数カナ文字セット 又は英小文字セット	IBM漢字コード

図2に示されるように日本語 PC/VM BONDでは、PC BOND プログラムと PCDISK ユーティリティでコード変換を行なう。PC BOND プログラムでは、パソコンとホストで交換するデータを変換する。PCDISK ユーティリティでは、CMS ファイルと DOS ファイルのデータについてのコード変換を行なう。ここでのコード変換方式の特徴を以下に示す。

- コード変換はすべて VM 側で行なう。
- PC BOND プログラムと PCDISK ユーティリティは、変換テーブルを共有する。変換テーブルは、PC BOND プログラムが持つ。
- PC BOND プログラムと PCDISK ユーティリティのコード変換は、特別な指定がない限り 3270PCの基本文字モード¹⁷⁾に従う。
- JISCIIと EBCDICのコード変換のほか、ユーザ定義文字等も扱えるようにするため、ユーザ独自の交換テーブルによるコード変換もサポートする。

3270PCファイル転送ユーティリティでは、コード変換をパソコン側で行なっているが、日本語 PC/VM BONDではコード変換をホスト側で行なうようにしたのは、以下の点を考慮したためである。

- 将来、パソコンどうしの結合を含めた連携処理が実現されるようになった時、JISCIIと EBCDICの間のコード変換は不要になる。
- ユーザー独自の交換テーブルを共有する場合を考えれば、交換テーブルを管理するのは、ホスト側の方がよい。

変換テーブルを PC BOND プログラムと PCDISK ユーティリティとで共有させた理由としては、2バイト・コードの変換テーブルのサイズが大きいことと、ユーザーが独自の交換テーブルを使用する場合を考えためである。この場合、日本語 PC/VM

BONDの変換テーブルを書き替えなくてはならないが、変換テーブルを共有することで、書き替えを一ヶ所にとどめておくことができる。

5. 日本語機能実現における諸問題

MML ソフトウェアの実現のためには、4. で見たようにリクエストとサーバ、それに、ホストとパソコンの文字コードの違いを吸収する機構が必要となる。とりわけ、2バイト・コードを処理するシステムでは、以下の理由により、扱いが難しくなる。

- EBCDIC には 2種類の 1バイト・コードが存在する。
- 1バイト・コードと 2バイト・コードとが混在する。
- 2バイト・コード・テーブルの一部はユーザ定義文字用に解放されている。
- 2バイト・コードの取り扱いがパソコンとホストでは異なる。EBCDIC では、2バイト・コードと1バイト・コードとを区別するために、一続きの2バイト・コードの、前後には制御文字 SO(Shift Out)と SI(Shift In)が入る。一方、JISCII では、2バイト・コードの識別は、コードそのものでできる。

以下、MML ソフトウェアを作る場合、特に2バイト・コードを扱う際に起こる問題とその解決方法を、米国版の PC/VM BOND を基に日本語 PC/VM BOND¹⁸⁾を開発した経験から述べる。

コード変換

4. で述べたように、日本語 PC/VM BOND では、コード変換をホスト側で行なった。ここでの問題とその解決方法には以下のものがあつた。

- ホスト側で 2種類の EBCDIC 1バイト・コードをどのようにサポートするか。
PC BOND プログラムと PCDISK ユーティリ

16 本稿では、DOS 側で使われる文字セットを 1バイト文字と2バイト文字とを区別せずに JISCII(Japanese Industrial Standard Code for Information Interchange)、VM 側のを EBCDIC(Extended Binary Coded Decimal Interchange Code)と呼ぶことにする。EBCDIC の1バイト・コードには、EBCDIC 英小文字セットと EBCDIC 英数カナ文字セットがあり、それぞれ英小文字と英数カナと呼ぶことにする。英小文字にはアルファベットの小文字があるが、半角カタカナ文字がない。一方、英数カナにはカタカナ文字はあるが、アルファベットの小文字がない。DOS 側ではアルファベットの小文字と半角カタカナ文字の両方を使うが、VM 側ではいずれかを選択しなければならない。

17 基本文字モードは、3270PCのカスタム時に英数カナか英小文字に決められる。

18 以後、基にした米国版の PC/VM BOND を「米国版」と呼ぶことにする。

ティの標準のコード変換は、PC BOND プログラムが起動時に 3270PC の基本文字モードを調べて決定するようにした。

- ユーザ独自の交換テーブルをどのようにサポートするか。
ユーザが変更した交換テーブルを仮想ディスク上の DOS ファイルとして持って、それを利用できるようなオプションを用意した。

2 バイト・コードのサポート

2 バイト・コードを使用するために、注意した点について述べる。

- コード変換により、データの長さが変わることがある。これは、EBCDICで 2 バイト文字を表すときに必要となる SO/SI が挿入されたり削除されるためである。
- PCDISK ユーティリティで、8 バイトより長いパラメータを使用できるようにした。DOS ファイル名などに対応するパラメータは、JISCI で 8 バイト以下の長さでも、EBCDIC に変換すると SO/SI が挿入されて 8 バイトより長くなることがあるためである。
- メッセージ・ウインドウに 2 バイト文字を表示するために、5550 BIOS¹⁹機能呼出しを使った。

日本語 3270PC の環境

日本語 PC/VM BOND では端末エミュレータに依存してプログラムを書いた。そのために、リクエスト(PC BOND プログラム)とサーバ(VM BOND プログラム)を作成する時に起こった問題を(2 バイト・コードの問題ではないが、)述べる。米国版は独自のプログラミング・インタフェースでホストとパソコンの通信を行っていた。日本語 PC/VM BOND では、3270PC の API を使用するように変更した。その際、VM BOND プログラムと 3270PC との間に、米国版のプログラミング・インタフェースをシミュレートするレイヤを置いた。

3270PC を使用したことで、端末インテリジェンスを利用できるようになった。米国版の端末エミュレータは、CUT (Control Unit Terminal) モードで IBM3274 クラスタ・コントローラとつながっている。一方、3270PC は、DFT (Distributed Function Terminal) モードでつながっており、より多くの機能を使える。それらの機能により、日本語 PC/

VM BOND では、データ交換に構造化フィールド²⁰を使うようにした。また、二つ以上の端末セッションで VM を使用できるときに、VM BOND プログラムと接続する端末セッションを選択できるようになった。

6. おわりに

本稿では、発展した形の MML ソフトウェアである日本語 PC/VM BOND と ECF が提供するサービスについて述べた。また、それらのサービスを実装する際に生ずる問題とそれに対する解決方法を特に 2 バイト・コードの扱が必要となる場合について、日本語 PC/VM BOND を例にとりて述べた。

ユーザがインタフェースの変更を感じずに使用できる資源をホストにまで拡張できるということは、ユーザがコンピュータを操作する上での負担を増やさないことを意味する。また、新たにハードウェアやソフトウェアを導入せずに資源やサービスを増やすことができるということで、非常に効率的である。さらに、SRPI によって、簡単にホストパソコンの分散処理プログラムが書けるようになることで、一層有効な資源の利用が期待できるであろう。

文献

- [相原 87] 相原,黒沢,金田,平賀: “日本語 PC/VM BOND (2)”. 情報処理学会第34回全国大会 7M-7 (March 1987) pp. 2153-2154.
- [Auerbach 85] Auerbach, J.: “File Request Transparency Between Heterogeneous Systems”. IBM Research Report RC11496 (November 1985).
- [BOND 86] “日本語 PC/VM BOND ユーザーズ・ガイド”. 日本アイ・ビー・エム株式会社 N: SB10-7793 (1986).
- [Goldstein 84] Goldstein, B.C., Heller, A.R., Moss, F.H., and Wladawsky-Berger, I.: “Directions in Cooperative Processing

19 Basic Input Output System

20 3270 データ・ストリームの一部で、スクリーン上のデータと独立であり、より高いデータ交換効率を期待できる。

- between Workstations and Hosts". IBM Systems Journal 23 (3) (1984) pp. 236-244.
- [萩野 85] 萩野和彦: "メインフレーム・WS連携システムにおける制御構造". 情報処理学会オペレーティング・システム研究会 OS28-1 (September 1985).
- [林 86-1] 林 秀幸: "ホストとワークステーションが連携するマイクロ・メインフレーム結合が急増". 日経コミュニケーション 1986年2月10日号 (February 1986) pp. 85-91.
- [林 86-2] 林 秀幸: "日本IBM, 拡張接続機能でホスト・パソコン連携の基盤確立". 日経コミュニケーション 1986年9月8日号 (September 1986) pp. 50-52.
- [平賀 87] 平賀, 黒沢, 金田, 相原: "日本語 PC/VM BOND (1)". 情報処理学会第34回全国大会 7M-6 (March 1987) pp. 2151-2152.
- [石田 86] 石田雅也: "本格的な普及へ向かうマイクロ・メインフレーム・リンク". 日経コンピュータ 1987年2月16日号 (February 1987) pp. 71-84.
- [黒沢 87-1] 黒沢隆: "日本語 PC/VM BOND". ELAN (January 1987) pp. 54-55.
- [黒沢 87-2] 黒沢, 相原, 平賀, 金田, 緒方: "日本語 PC/VM BOND: マイクロ・メインフレーム結合の実現例". 情報処理学会マルチ・メディア通信と分散処理研究会報告 87-MDP-32 (March 1987).
- [KURO87] Kurosawa, T.: "Japanese PC/VM Bond: An Example of Cooperative Processing between Personal Computer and Host Computer". Denshi Tokyo 25 (1986) pp. 40-45.
- [Parr 85] Parr, F.N., Auerbach, J.S., and Goldstein, B.C.: "Distributed Processing Involving Personal Computers and Mainframe Hosts". IEEE Journal on Selected Areas in Communications SAC-3 (3) (May 1985) pp. 479-489.
- [真田 86] 真田英彦: "MMLを実現するためのホストコンピュータとパソコンの機能: 4段階に分けて結合のステップをみる". コンピュータ&ネットワークLAN (November 1986) pp. 37-40.
- [関山 86] 関山茂寛: "IBMが実現するワークステーションとホスト・システムのコネクティビティ". コンピュータ&ネットワークLAN (November 1986) pp. 41-48.