

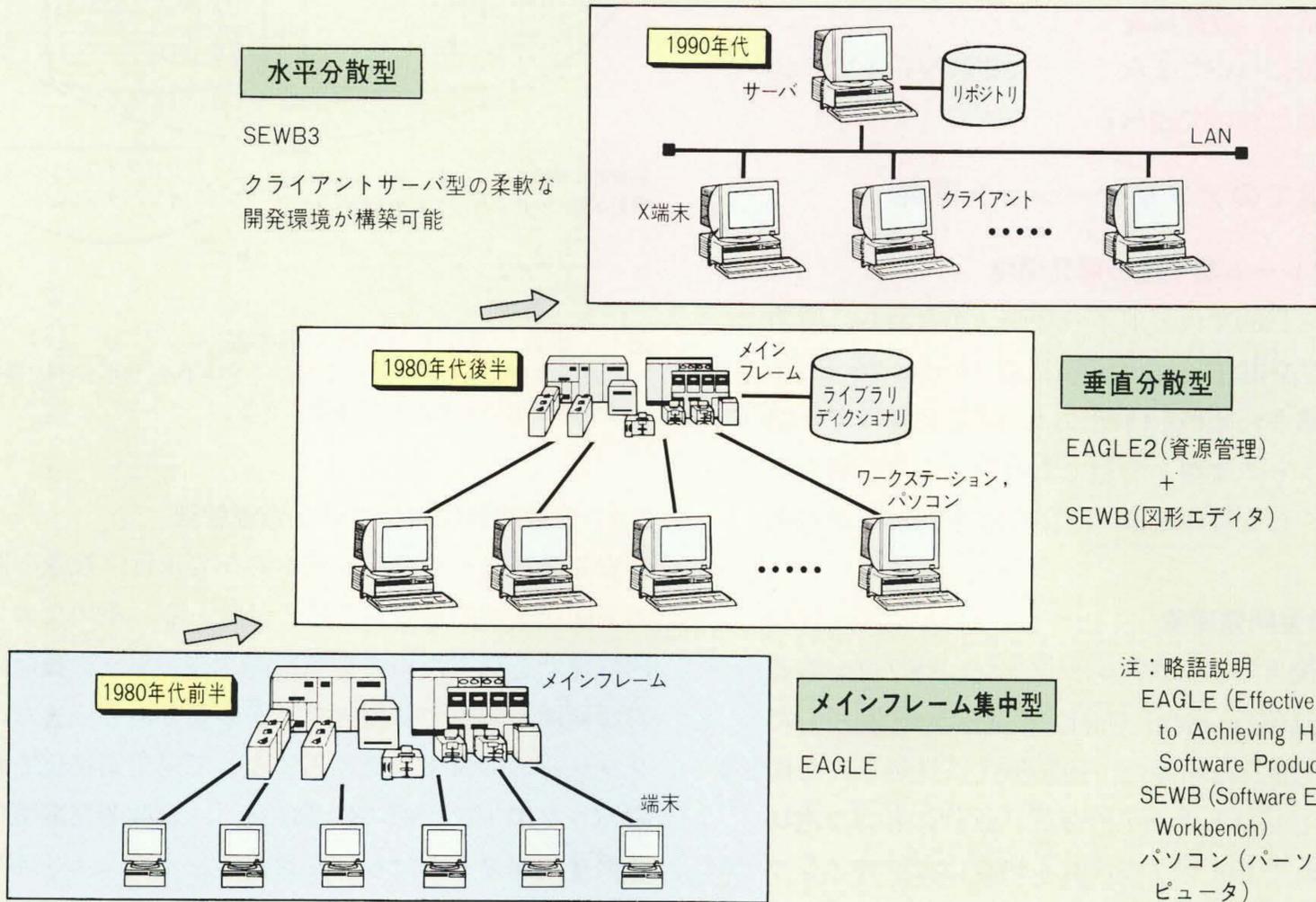
分散コンピューティング環境における アプリケーション開発支援

Application Software Development Environment in Distributed Computing Era

吉野松樹* Matsuki Yoshino

秋山美登* Yoshinori Akiyama

西尾高典* Takanori Nishio



アプリケーション開発環境の変遷 メインフレーム集中型、垂直分散型を経てアプリケーション開発環境は水平分散型へ移行しつつある。

近年の企業を取り巻く経営状況の変化に伴うリストラクチャリング、ダウンサイジングといった動きによって、情報処理の形態も従来の集中型から分散型へと大きく変革を迫られつつある。

この変革を技術的に支えているのが、ワークステーション、パソコンなどのハードウェアの急激な進歩と低価格化、またそれによって現実的なものとなったグラフィカルユーザーインターフェース、クライアントサーバコンピューティング、オブジェクト指向パラダイムなどのソフトウェア技術の普及である。

アプリケーションを開発する環境も、かつてのメ

インフレーム集中型の開発環境から、メインフレームとワークステーション、パソコンが協力する垂直分散型を経て、ネットワークで接続されたワークステーション、パソコンが協調する水平分散型に移行しつつある。

また、分散コンピューティング環境で実行されるアプリケーションを開発するためには、開発方法論、開発支援ツール、開発支援環境それぞれが従来とは異なるアプローチが求められている。

両者の観点から、日立製作所の取組みの状況を中心に、現在までの状況と今後の動向を紹介する。

* 日立製作所 ソフトウェア開発本部

1 はじめに

マイクロプロセッサの進歩やLANの普及などのハードウェア環境の変化に対応して、アプリケーションソフトウェアの開発環境、開発の方法も変化しつつある。開発支援環境もこの変化に対応し、47ページの図に示すような変遷をたどっている。ここでは、開発支援環境の変遷について2章で、分散環境で実行されるアプリケーションの開発支援について3章で、それぞれ現在までの状況と今後の動向について述べる。

2 分散環境でのアプリケーション開発

2.1 メインフレーム集中型の開発環境

1970年代から1980年代の前半の時期、すなわち、開発環境あるいはCASE(Computer Aided Software Engineering)が認識され始めた時期には、開発環境も実行環境もメインフレーム集中の形態であった。この時期の開発環境として、日立製作所は、EAGLE¹⁾を1983年から出荷している。

2.2 垂直分散型開発環境

1980年代の後半、ワークステーション、パソコンの普及に従い、GUI(Graphical User Interface)を利用したCASEツールが普及し始めた。1970年代に提唱された構造化設計技法でソフトウェアの分析、設計に有力な道具として考案されたさまざまな図式を扱うには、メインフレームのキャラクタベースのユーザーインタフェースでは不十分であり、GUIが必要だからである。この時期の開発環境は、DFD²⁾(Data Flow Diagram)、PAD³⁾(Problem Analysis Diagram)などの図式の編集を行うワークステーション、パソコンと、編集された設計情報を管理しその設計情報からアプリケーションプログラムの生成を行うメインフレームという、垂直型の分散形態をとっていた。

1986年から出荷しているSEWB⁴⁾とEAGLE 2との組み合わせの形態は、この形態の開発支援環境である。

2.3 水平分散型開発環境

1990年代に入り、ネットワーク技術の普及、ワークステーションの信頼性、ディスク容量の増大などによって、設計情報をワークステーションで管理し、ネットワークで接続されたワークステーションあるいはパソコンからその情報を利用するという、水平分散の開発環境が可能となった。

1992年に出荷を開始したSEWB 3⁵⁾は水平分散開発環境を実現しており、ユーザーが開発の状況に合わせて柔軟に開発環境を構築できるようにしている。

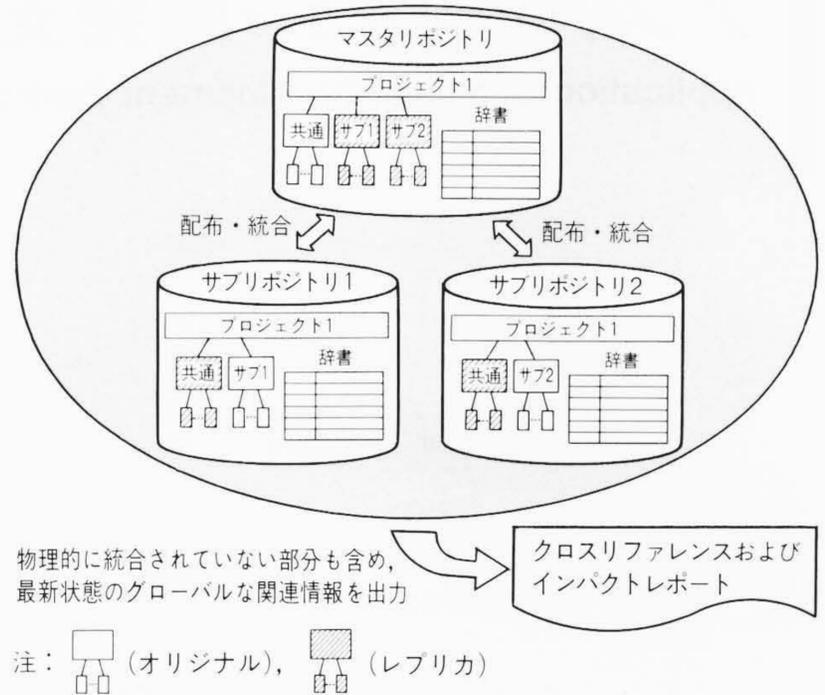


図1 複数リポジトリ運用支援機能

この機能により、物理的に複数に分割されたりリポジトリを論理的に一つのリポジトリとして運用できる。

2.4 分散開発環境における資源管理

資源管理に対する最も基本的な要求は、複数の開発者の共同作業が円滑に行えることである。そのためには、同時更新を防ぐための排他制御、バージョン機能、変更影響範囲を調査する機能などが必要である。また、メインフレームではセンタ運用によって定型業務化されているバックアップなどの作業に対し、分散開発環境でも同様の運用ができなければならない。

SEWB 3ではこのような資源管理の機能を、ワークステーションリポジトリによって実現している。

開発規模が非常に大きい、あるいは開発拠点が複数に分かれているなどの事情により、一つのプロジェクトで複数のリポジトリを使用する場合には、複数のリポジトリ間での情報の配布・統合、整合性の確保などを支援する機能が必要となる。SEWB 3では、図1に示すような複数リポジトリ運用支援機能により、物理的には複数のリポジトリを、論理的には一つのリポジトリとして運用できる(開発中)。

3 分散環境で稼動するアプリケーションの開発環境

3.1 クライアントサーバコンピューティング

クライアントサーバコンピューティングは、サービスを提供するサーバと、そのサービスを楽しむクライアントが協調して一連の作業を遂行するという考え方である。

サーバとクライアントというのは、どのサービスに着

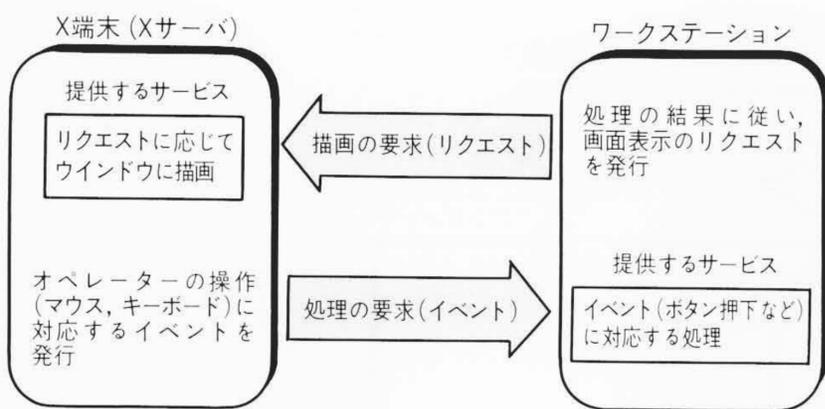


図2 クライアントサーバの例
サーバとクライアントの関係は、着目するサービスによって変化する相対的な概念である。

目するかによって変化する相対的な概念である。同じシステムの中でも一つのマシンがあるサービスに関してはサーバであると同時に、別のサービスに関してはクライアントにもなり得る。例えば、図2に示すように、ワークステーションにX端末を接続している場合、ウィンドウに描画するサービスは、X端末がワークステーションに対して提供しているサービスなので、X端末がサーバであり、ワークステーションがクライアントである。X端末がXサーバとも呼ばれるゆえんである。逆にウィンドウに描画する情報を作っているのはワークステーションなので、このサービスに着目すればワークステーションがサーバであり、X端末はクライアントである。

サーバが提供するサービスとしては、CPUの計算能力、データベース機能、他システムとの接続(ゲートウェイ機能)、印刷機能などさまざまなものが考えられる。

3.2 分散環境で稼動するアプリケーションの形態

データベースを検索して検索結果を加工して表示するという処理を、図3に示すように、① ウィンドウおよびキーボード、マウスの制御、② 入力データの妥当性チェック、③ 入力データを解析してデータベースへの問い合わせの発行、④ 問い合わせに従ってデータベースを検索する処理、⑤ 検索結果の編集・加工処理の五つの処理コンポーネントに分割する。

データベースサーバとクライアントの間での処理分担の形態を考えると、図3中に示すような(a)から(e)までのさまざまな形態が考えられる。

図3(a)の形態をワークステーションまたはパソコンをクライアントとして実現する例として、日立拡張表示印刷システムXP(Hitachi Extended Presentation Service System)がある。SEWB3あるいは第4世代言語EAGLE/4GLを利用すると、XPシステムを利用した(a)あるいは(e)の形態のアプリケーションを従来のメインフレームの

アプリケーション開発と同様の考え方で開発することができる。また、分散トランザクション処理機能BeTRANを利用したトランザクション処理のアプリケーションも同様の考え方で開発できる。この方法は、クライアントサーバコンピューティングのすべての形態には対応していないが、既存の資産、ノウハウを活用してダウンサイジングを行うことができるという大きなメリットがある。

(e)の形態は、データベース管理システムの分散機能によるクライアントサーバシステムである。

(a)から(e)のどの形態をとるのが最適であるかは、分散コンピューティングのインフラストラクチャとしてどのようなものを利用するか、サーバ、クライアントのそれぞれのハードウェアの能力、ネットワークの負荷、開発効率などさまざまな要因に依存する。

日立製作所が提供するオブジェクト指向アプリケーション開発環境Object IQの分散アプリケーション開発機能を利用すれば、DCE(Distributed Computing Environment)環境で実行される(a)から(e)のすべての形態のアプリ

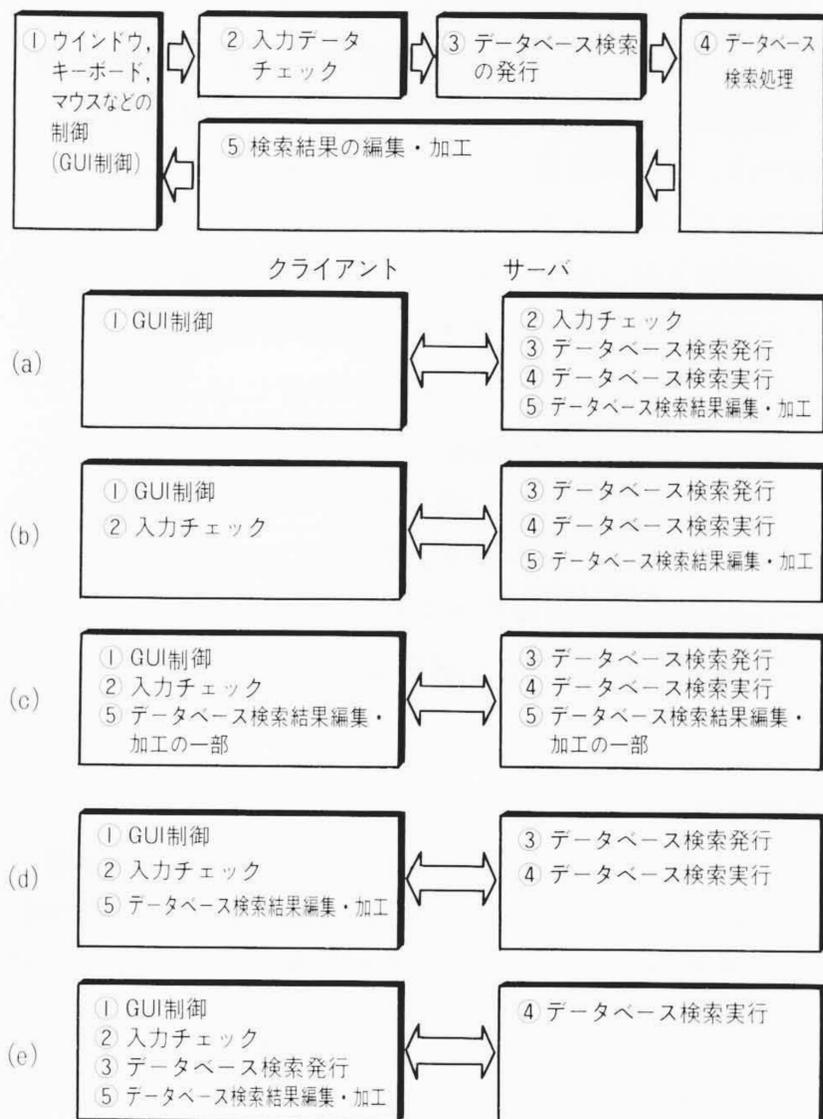


図3 データベース検索アプリケーションを例にしたクライアントサーバ処理形態

同一のアプリケーションでも、クライアントとサーバの役割分担にはさまざまな形態が考えられる。

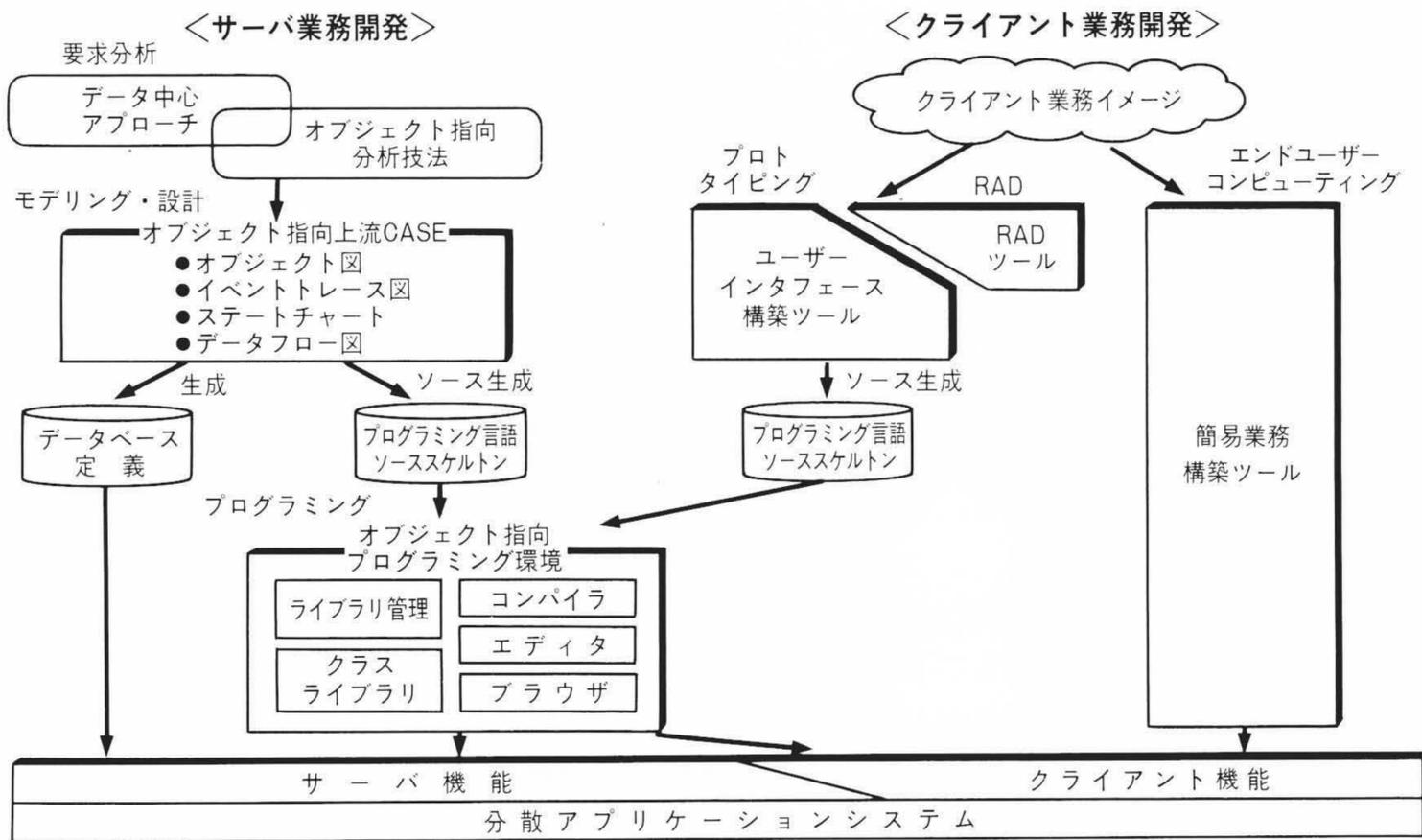


図4 分散コンピューティング環境をターゲットとした開発環境アーキテクチャ
分散コンピューティング環境をターゲットとする開発では、オブジェクト指向、RADなどの新しい手法が期待できる。

リケーションの開発が可能である。Object IQでは、クライアントとサーバの間の通信の詳細を意識せずに分散アプリケーションを開発することができる。

3.3 分散環境で稼動するアプリケーションの開発支援

分散環境で稼動するアプリケーションの開発の上流工程では、サーバ機能とクライアント機能の切り分けが重要である。サーバをオブジェクトとしてとらえ、サーバが提供するサービスをオブジェクトが他のオブジェクトに公開するメソッドと考えれば、サーバとクライアントの機能の切り分けとサーバ機能の設計には、オブジェクト指向分析・設計⁶⁾が適用できる。

クライアント機能は、エンドユーザーが直接操作するGUI中心の処理が多いので、RAD⁷⁾(Rapid Application Development) ツールなどを利用したエンドユーザーコンピューティング⁸⁾による開発効率向上が期待されている。

このようにサーバ、クライアントそれぞれに異なる開発支援ツールの適用が多くなるが、開発の過程で作成される資源は一元管理できる開発環境が必要である。

分散コンピューティング環境を対象とした開発環境のアーキテクチャの構想の一例を図4に示す。

4 おわりに

2章で述べたように、開発支援環境の分散化はSEWB3によって実現されている。3章で述べた実行環境の分散化に対応する開発支援環境は、実行の形態が多様であることや、従来の構造化分析・設計技法だけでは十分対応できない要因も多く、解決すべき課題が多く残されている。日立製作所は、今後も情報処理形態の主流になると予想される分散コンピューティングに対応した開発方法論、および開発環境の整備を図っていく⁹⁾。

参考文献

- 1) 曾根原, 外: EAGLE大規模オンライン向け開発支援の強化, 情報処理学会第33回全国大会論文集, 613~614(1986)
- 2) Gane, C., et al.: Structured Analysis: Tools and Techniques, Prentice-Hall(1979)
- 3) 二村: プログラム技法-PADによる構造化プログラミング, オーム社(1984)
- 4) 葉木, 外: "SEWB"の開発思想と機能, 日立評論, 70, 2, 101~108(昭63-2)
- 5) 吉野, 外: ソフトウェア開発支援ツール"SEWB 3, EAGLE/4GL"の機能と特長, 日立評論, 75, 11, 727~734(平5-11)
- 6) Rambaugh, J. et al.: Object-Oriented Modeling and Design, Prentice-Hall(1991)
- 7) Martin, J.: Rapid Application Development, Mac-Milan (1991)
- 8) 野口, 外: エンドユーザーコンピューティングの現状と将来展望, 日立評論, 75, 9, 570~576(平5-9)
- 9) 橋本, 外: オープン化, マルチベンダ化にこたえるクライアントサーバシステム構築支援技術, 日立評論, 75, 11, 751~754(平5-11)