

経営学に影響を与える情報科学発展の軌跡(3)

…産業人となる学生に適切な情報化教育をしているか？…

和多田 作一郎

I 概 要

ここ数年前からクライアント・サーバー型のパソコン LAN（クライアントサーバー・システム，CS システムと略称）が，企業の情報インフラとして急浮上してきた。ダウンサイジング機器（パソコンやワークステーション等）の活用，オープンシステムの構築そしてエンド・ユーザー・コンピューティング（EUC と略称）という 3 つのキーワードを実現するに最適なシステムであることが認識されたからである。

現在，筆者は企業のトータルシステムの構築に向けての LAN 導入の指導をしている。後述する二階層クライアント・サーバー型 LAN と称せられる病院の医事システムと，三階層 LAN あるいは大規模 LAN（あるいは広域 LAN）と称せられる物流業者の物流システムの構築である。この経験から考えると，日本の LAN 技術は米国に比べて10年程遅れていると云われているが，それを実感した。さらに，システム・インテグレータ（以下 SI と略称する）と称せられるシステム構築企業の技術者の LAN 構築の知識不足には深刻なものがある。

さらに当校にお世話になってから，各大学の経営学部のカリキュラムを調査した。またこの度の LAN 導入指導の経験から大部分の大学の情報化教育のカリキュラムは，汎用コンピュータ時代の教育を踏襲して，次世代を生き

抜く経営・管理者になる人に対する適切な教育か、と云う点になるといろいろと問題があるのではないかと考えている。

この業界は将に、時進分歩の勢いで技術が進歩している。学生が入学をして、卒業するまで4年間かかり、さらに4年後に一人前の企業人となるためには、少なくとも10年後を見据えた教育をしなければならない。このように考えると、現在このような教育をされていていいのか、と寒心に耐えない。

(社情報処理学会が1990年に創立30周年を迎え、その翌年の1991年1月号で「30年後の情報処理」という特集を発刊した。今年はそれからまだ4年後にしかないが、あまり当たっているとは思えない。そのなかの一つの項目に、次のような表現がある。「コンピュータはすでに社会のインフラストラクチャになっているから、これに合わせて社会制度自身も硬直化してしまう。つまり、社会制度を既存のソフトウェアに合わせてしまうというようになる。(以上4節の一部)」(中略)「過去30年間、Fortranは不滅であった。CもUnixも今後30年間は不滅かも知れない。これは4. で述べたことと関連するが、ソフトウェアというのは一つの文化であるから、目まぐるしく変化するというより、蓄積していく性質のものである。従来の文化と異なり、情報流通が瞬時かつワイドに起こる時代だから、地域に文化が付随するという図式は成り立たない。」(以上7節の一部)¹⁾ コンピュータに関するかぎり前提条件を明確にして論議しないと、誤った解釈をするが、この考えは間違っているとはしか考えられない。少なくともエンド・ユーザー・コンピューティングという立場で考えると、「コボル軍団崩壊」と云う本がでているが、現在コボル言語を使う人が非常に減ってきたし、パソコンのエンド・ユーザーは、作表ソフト(例えばExcel, Lotus123等)を利用して簡単に非定型処理を可能にしている。これからの経営情報インフラとなるクライアント・サーバー型のLAN構築にかぎり、これまでの手続き中心のコボルによるシステム構築の知識が、これからデータ中心のシステム構築時代には、頭の切り替えを困難にしていると感じた。

東京都の豊島区にある光通信(株)は第二電電の代理店でオープンシステムの

構築に先進的な SI である。その情報システム部に、「設計しない，コーディングしない，テストしない」という標語が貼り出されている。詳しく後述するが，プログラムを作らないで，流通ソフトを組合せて，情報システム部門でなく，エンドユーザー自身がシステム構築を主導する時代が到来したのである。今から 4 年前の 1991 年に，コンピュータの巨人といわれた IBM が，その研究費総額の三分の一を汎用コンピュータの研究に振り向けていたという事実に注目していただきたい。これが同社の急激なる業績悪化になった最大の原因で，クライアント・サーバー型のパソコン LAN がこれだけ急速に普及して大部分の汎用コンピュータの換わりをすることに気づかなかったからである。

この業界は現在，パラダイム・シフトとでもいうべき急速なる構造変革を受けている。このパラダイム・シフトの方向を見定めて，十年後に役立つ経営学部の卒業生の役立つようなカリキュラム再編成の参考になるように，現在コンピュータ活用の現場で起こっていることを説明する。後半は，将来の情報処理技術の変化を展望して，どのような情報化教育を施したらよいかの私見を述べる。

大学冬の時代を迎えて，当校が他の大学と差別化して，有能な企業家精神に溢れた企業人を養成して発展していただくことを祈念して拙い筆をとることにする。

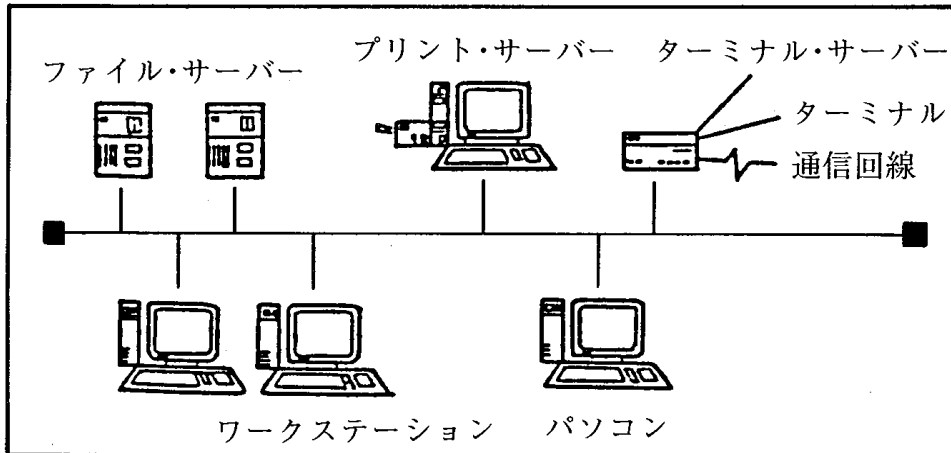
前半では，コンピュータ関係の専門用語を可能なかぎり使わないし，使う場合は用語に解説を加えて説明をするので，なるべく多くの関係者にお読みいただきご批判をお願いしたい。

用語解説

・クライアント・サーバー・システム

次頁に代表的なクライアント・サーバー型のパソコン LAN を示す。クライアント・サーバー・システムというのは「あるクライアント（パソコン）がサーバーに対して処理を要求して，その結果をもたらすという形で，複数のクライアントが同時に処理を要求しても応答してくれるシステム」と定義することができる。このシステムのメリットは LAN を構成する配線が大容量のデータ伝送を可能にし，しかも一本の配線をサーバー，各クライアントに配線しているので配線の集約効果がで

る。さらに大容量のデータの伝送が可能である。サーバー資源であるファイル（データベース）やプリンター等を各クライアントで共用できる等の多くのメリットがある。



・オープンシステム

「オープンシステム」という用語は、コンピュータ活用に革命とも言える大きな転換をもたらすパラダイムと考えるべきである。IBMの汎用コンピュータは「クローズド」なシステムであった。他社の機種との接続、ソフトの交換性を考慮しないことを基本戦略とした開発思想であった。これに比べて、DEC社のミニコンはさまざまな機種製品をラインナップとして揃えるときにOS（基本ソフト）をすべて統一することにより、それらすべての相互接続をすべて可能とした。狭義のオープンシステムはアプリケーション・インターフェイスを公開することによって複数メーカーから提供される異機種のコンピュータを接続して、データ交換を可能ならしめることをいう。現在普及しているパソコン通信はこの狭義のオープンシステムである。

広義のオープンシステムはOSをも含んだシステム環境全体に関するもので「UNIX」は稼動アプリケーションが豊富でオープンシステムを構築するためのプラットフォームといえることができる。最近普及しているパソコンのWindowsも広義のオープンシステムを可能ならしめるプラットフォームである。オープンシステム環境が整備されると、いろいろなベンダーのハード・ソフト製品を活用することのできるマルチベンダー環境が実現し、飛躍的に使いやすく、安価にシステムを活用できるようになる。

・エンド・ユーザー・コンピューティング（EUC）

従来のシステム構築は外部の業者や情報システム部門がシステムを構築してくれたが、これはコンピュータの利用者自身がシステムの企画、開発、運用することという。最近ではエンド・ユーザー開発（End User Development: EUD）という言葉もでてきた。これはユーザー自身が定型的な処理の業務系システムを構築する場合をいう。本稿では、非定型処理の情報系のシステムを活用して、作表ソフトで、データベースの情報を検索して非定型処理をする場合とこの定型的な業務系のエンド・

ユーザー開発を含めて、エンド・ユーザー・コンピュータということにする。今後このような方向に確実に進展するからである。

- ・システム・インテグレーター (SI)

情報通信システムの構築を支援する業者で、ユーザーの抱える業務上の問題にあわせて、総合的に経営情報システムを構築し、保守、運用に必要な業務を一貫的に行う。

2 コンピュータ活用の流れが変わる

(1) これからの流れを理解するキーワードは「情報の共有化」

当初の企業のコンピュータ活用は、EDPS という用語が示すように「データ処理システム」としての活用が主で、省力効果の追及が目的であった。その後、ネットワーク化が進展して、リアルタイム処理、分散処理を可能として、組織の効率化、つまり経営の生産性向上のツールとしての活用が始まった。同時に OA 化が進展して、オフィスの生産性向上は、ワープロやスタンダードアロン・パソコンを導入して、個人業務をいかに効率化するかという視点での検討もはじまった。

その後、右肩上がりの高度成長経済に続くバブル景気の崩壊とともに大競争時代が到来した。厳しい経済環境のもとに、一刻を争う時間重視の経営が求められ、コストダウン競争が激しくなるばかりで、これまでと同じやり方の経営では、生き残ることのできない新業態創造の時代に突入した。各企業では新業態創造のための新しい経営のやり方に知恵を絞り、新しい情報システムの構築による生産性の向上にしのぎを削るようになってきた。

この新しい情報システム構築の流れの方向を理解するキーワードは「情報の共有化」である。情報の共有化とは、経営活動に必要な「ヒト・モノ・カネ」などに関する「みえる経営資源」に関する情報を社内の経営階層や部門を越えて、誰でも自由に利用できるようにすることである。それ以上に必要なことは、個人のノウハウとして頭の中に蓄積されてきた知識・情報や、企業の各部署に蓄積されている知識・情報という「みえない経営資源」を企業全体の資産として活用することを可能にすることである。経営トップの有す

る情報と営業の第一線のセールスマンが有する情報が共有できれば、新しい経営スタイルが生まれる。

(2) 「情報の共有化」は経営組織を変える

H. A. サイモンが、経営組織は意思決定のシステムであり、経営管理とは意思決定の連続である、という考えを発表した。これにより、伝統的組織論の行動システムから情報処理システムとしての組織のパラダイムが登場することになる。

その後、情報通信技術の発展は、広範なるマーケットデータをリアルタイムに把握することが可能になった。この広範なるマーケットデータを経営に役立つ情報に変換するにも、その情報に基づいて行動するにも、モノの大量生産時代の組織では不向きなことが次第に分かってきた。規模の経済性を追及する目的で作られた分業組織では、組織の壁が情報の流れを阻害し、大競争時代の現在では多品種少量生産向きの経営組織としては不適當となった。これを克服して環境の変化により柔軟に適応できる新しい組織形態が模索されることになる。

一方、1980年に日本の企業に追い詰められたアメリカの「エクセレント・カンパニー」の業績の低下するなかで、M & A（企業の合併）等を通じて各種経営資源を統廃合するリストラクチャリングが注目された。しかし、それによって競争優位を維持し増加させた企業は少なかった。競争優位を獲得するには、経営資源を切り捨てたり、統廃合をすることによって得られるものではないという、反省がおきた。

この反省から、顧客の満足を追及して、顧客を理解して、顧客の側からビジネスの流れを根本的、抜本的に見直して、それまでの機能本位に作られた組織を再構築して、より効率的な経営の仕組みを作るというリエンジニアリング（正確には、ビジネス・プロセス・リエンジニアリング、略称BPR）が注目された。日産自動車が生産工場を閉鎖するほど業績悪化に苦しむなかで、アメリカの自動車産業がよみがえった。ビッグスリーの一角であるクライスラー社が、長年業績不振に悩んだが、奇跡の業績回復を成し遂げた。こ

れがリエンジニアリングの一つの成功例である。

これまでの組織図に表示されていない「ビジネス・プロセス」つまり業務の流れに注目して、それを根本的に見直して、顧客満足を追及するモノの流れ（サービスの流れと考えてもよい）にやり変えることによって、アメリカの企業は劇的・画期的な業績の向上を可能ならしめた。

このほか、米国の企業は多くの新しい経営管理の手法を開発している。後述するコンカレント・エンジニアリングや仮想企業である。これらのすべては、情報の共有を可能な情報通信システムを活用して、縦割り・階層型の伝統的な部門体制で「処理する」のではなく、プロジェクト制・チーム制などの情報横断的な組織体制で「価値を創造する」形態に変わってきている。

何よりも新鮮な考えは「組織図に示されていないビジネス・プロセスに注目して、これを顧客サイドから見直す」という考え方である。さらに重要な点は、これによって業績を劇的・画期的に向上させたことである。

「世の中は生きている、理屈は死んでいる」（勝海舟）という格言がある。時代の変化に応じて生産性向上可能な経営システムに変革しないと生き延びることはできない。高度成長経済時代と現在の大競争時代の経営はまったく異なることを理解しなければならない。

(3) 経営組織を包含し一体化した経営情報システムの構築

急テンポに変化する環境変化に適応できる企業だけが生き残れる。これまでのように人間のやっていた仕事をコンピュータにやらせるだけでは生き残れない。

これまでの企業の組織図には、経営管理の階層、部署が示されているが、人間の代わりに情報を処理し、伝達する情報通信システムは表示されていない。ヒト・モノ・カネという「みえる経営資源」だけを考えた経営学を古典的経営学というならば、これからの時代は「みえる経営管理」に情報、知識、ノウハウ、時間といった「みえざる経営資源」を含めた新しい経営学を創造すべき時代である。

これからの企業にとり「人間と情報通信技術（IT と略称）」は、経営を支

えるに不可欠な両輪であり、これからの経営組織は、人間（これまでの組織）と情報通信システムを一体と考えた新しい組織形態の創造の時代である。

このような経営情報システムの構築には、図1に示すように、クライアント・サーバー・システムが好適なことが理解されるであろう。従業員各個人にパソコンを持たして、クライアントがデータを要求して、サーバーがその処理を実行して、応答するシステムである。これによって、情報の共有を目的とした、自律分散型のシステムの構築が可能となる。

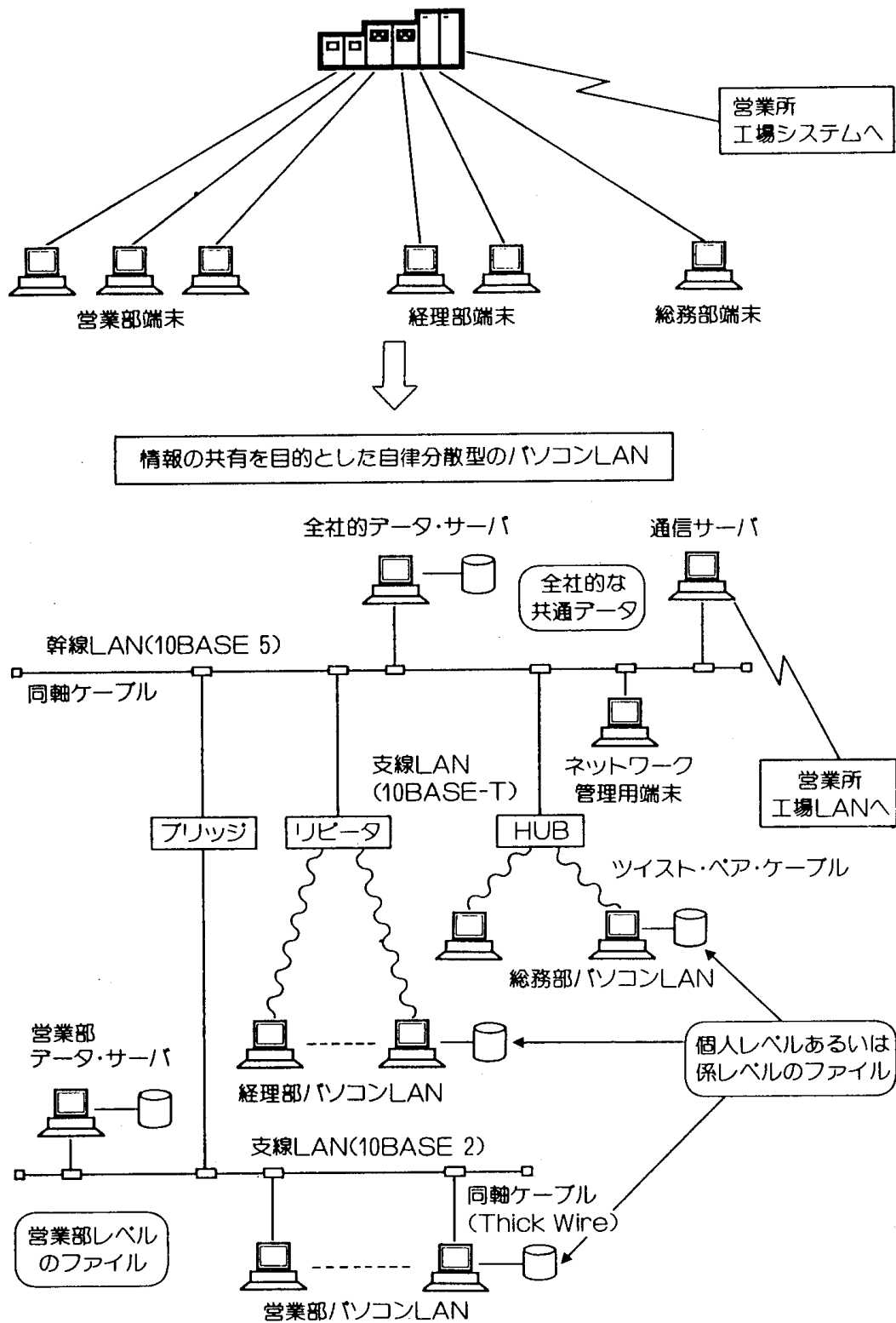
図の上に示すホスト中心の分散システムでは、主として定型事務処理を目的としていた。クライアント・サーバー・システムはこのような定型事務処理も可能であるが、これからは、組織や個人の生産性をあげるための「創造的な仕事」が求められる。定型事務処理の目的は、「今日」の経営の実体を把握するのに重要であるが、「創造的な仕事」は経営の「明日」を決める的確な意思決定をするのに必要である。ある人の考えた素晴らしいアイデアを、周囲の人に伝えて、実行プランにするための共同作業のための「グループウェア」の活用も可能であるし、これからは、マルチメディア時代を迎えて、格安にパソコン端末でカラーの動画を見ることが可能となる。このことは、遠方の人とテレビ会議ができるばかりでなく、グループウェアを活用して共同作業が可能になる、ということの意味する。

本節で述べたこと以外の LAN 活用の C/S システムのメリットを下記する。

a. 経営組織が環境の変化にフレキシブルになる

図の上に示すホスト中心のシステムは、EDPS 部門が構築したもので、各部署に適合したものとなっていない。これに比べて、パソコン LAN システムを構築すると組織にぴったり適合したシステムの構築ができる。組織の変更にも LAN 間接続機器によって柔軟に対応することができる。これからの経営において、経営組織と情報システムは表裏一体に完全にマッチしていなければならない。

図1 ホスト中心の分散処理から
パソコン LAN による情報インフラへの転換



b. 経営の階層、部署に応じた情報の伝達と共有が可能

LAN は自律分散型システムの構築が可能である。多様化する時代に対応するためには、経営方針に基づいて各階層・部署で自主的・能動的に仕事を進めることが重要である。

このために必要なのは、必要な人に必要な情報を伝達し、情報を共有して経営方針に基づいた管理をすることである。パソコン LAN では、職場の個人が、一人一台のパソコンを活用して、時間と距離を克服して、リアルタイムに経営の実体が把握できる。

クライアント・サーバー・システムはまさにこのために最適なシステムである。

以上述べたことは、経営管理面の効果であるが、コンピュータ・システム面の効果として次が重要である。

c. CPU の処理能力をデータ処理能力に応じて増大

CPU（コンピュータの頭脳に相当する中央処理装置）の処理能力は、図の上に示すホスト中心の分散処理では、端末からのホストの活用台数が増えるにつれて低下する。

これにたいして LAN の場合は、端末の台数が増加するほど処理能力が増加する。事務処理量が増加した場合、ホスト・コンピュータを増設するのではなく、パソコンを増設することで対応することができるこのため非常に格安なシステム構築ができる。

この外、LAN の端末ネットワーク効果（ホスト・コンピュータが各端末にそれぞれ個々に線がでているが、LAN の場合は一本のセグメントに接続する）である配線の集約効果がある。さらに、各端末（クライアント）がサーバー資源を共有できるという資源共有型のシステムの構築ができる。

(4) 「情報の共有化」が生む新しい経営・管理手法

1960年から80年にかけて、わずか20年間で日本の製造生産性が6倍成長したのに、アメリカのそれはやっと2倍であった。鉄鋼、テレビ、エレクトロニクスといったアメリカの基幹産業は軒並み日本にマーケットを奪われた。

そのころ日本はアメリカより小型自動車を約2千ドルも安く作れた。これにより1980年、日本は世界一の自動車生産国のタイトルをアメリカから奪った。85年には日本は世界一の債権大国となり、87年には日本の一人当たりGNP（国民総生産）が世界一になった。

しかし、日本の企業はその後「おごりの経営」を続けたために、そして円高によって世界一高賃金の国になり五期連続の減収減益を余儀なくされる企業が多く、現在、平成不況の深刻さのなかにあえいでいる。

このようななかで、アメリカの産業が見事に再生をした。クライスラーは世界一安くて品質の良い自動車を作った。フォードもGMも変わった。日本の製造業の改善という考えを、「作業の改善、システムの改善、そしてシステムの戦略支援」にまで広げた「カイゼン」（日本語を使う）という概念にまで高めた新しい経営手法を開発したのである。これまで賃金だけを要求して働かないと噂されたアメリカの労働者を、年功序列・終身雇用の日本的経営風土に育った日本の従業員以上に、経営に協力する労働者に再教育したのである。

トヨタ自動車の「ジャスト・イン・タイムの生産」（JIT生産）を学び、それ以上に生産性の高い「リーン生産方式」を創造した。このためのリエンジニアリング（BPR）やコンカレント・エンジニアリング（CEと略称、同時平行開発・生産方式）等の「情報の共有化」のためにコンピュータを活用した経営手法を開発した。

製造会社と系列企業間をネットワーク化して、提携する仮想企業という手法も開発した。流通業界では、製造・販売の提携によるクイック・レスポンス生産・販売システムが急速に普及して流通構造を変えている。日本に進出して高業績なトイザラス等の新業態の小売業が次々と創造されている。

米国でパソコンを通じて銀行を利用する「パソコン・バイキング」が10月26日から本格稼働を始めた。（1995年10月28日、日本経済新聞夕刊）ケミカル・バイキングなど大手米銀19銀行が通信ソフト会社インチュイット開発の家計・財務管理ソフト「クイッケン」と銀行口座を結んだオンライン業務を開

始した。情報通信ネットワークのインターネット上で使用する電子マネー（eキャッシュ）の発行も23日始まった。支店が一つもなく、インターネットだけで利用できる銀行も誕生している。いずれも、パソコンで公共料金やクレジットカードなどの支払ができ、「銀行窓口はパソコンの画面」という時代の幕開けである。

これからの経営管理の手法は劇的・画期的な変革を余儀なくされる。

これらの新しい経営手法、新業態を理解するには、より便利な社会の建設に向けての人間とコンピュータ、コンピュータと企業間そして社会のオープンシステム技術の理解である。これによって、「情報の共有」のためのコンピュータ活用と、人間の創造性を発揮する「人間と情報通信システム」一体の新組織を創造することができる。

(5) 「情報の共有」に向けての技術基盤の確立

「情報の共有」を支える情報技術としては、強力で柔軟な機能を持つデータベースと、情報通信のためのネットワークが必要である。データベースの蓄積・処理した情報を企業内で活用するためのクライアント・サーバー・システムの構築と、この蓄積情報をスムーズに利用するための、グループウェアや電子メールなどの活用である。さらに、異企業間に必要な情報を何時でも伝達して「コンピュータに理解」させないと、前述の仮想企業やクイック・レスポンス・生産・販売システムの構築ができない。ここで、「コンピュータに理解」という言葉を使ったが、このためにコンピュータ間に共通の標準言語を活用して、通信をすることが必要である。これが現在急速に脚光を浴びているCALSである。このような技術基盤が日本は非常に遅れている。

しかし、集まった情報をデータベースなどに蓄積して、誰でもアクセスできるようにしたから情報の共有化ができる、というものではない。情報の履歴を的確に理解し、アクセスする人の業務内容に応じて、データを利用する権限を制限する必要も生じる。さらに米国で実用化した電子マネー「eキャッシュ」の普及には、セキュリティ問題が非常に重要である。

もっと重要なのが、エンド・ユーザー自らが情報を活用するマインドを持

つことが必要である。このような情報の使いこなしの能力が「情報リテラシー」である。

本稿では、この目的の技術基盤であるオープンシステム構築に向けてのクライアント・サーバー・システムの現状と将来展望を述べ、さらに情報ネットワーク技術の展望を述べる。

用語解説

- ・グループウェア（ワークフローとの違い）

グループや組織の活動を支援するソフトの総称で、ロータス・ノーツがよく使われている。それには電子メール、スケジュール管理、情報の蓄積・共有そしてソフトの自動実行の4つの機能がある。グループの業務を遂行しながら、自然に業務上の情報やノウハウが蓄積できるようになっているので、企業の知識蓄積のツールとして脚光を浴びている。

「ワークフロー」という複数の人間の業務を支援するソフトがあるが、ワークフローを支援する機能を有するグループウェアが多い。しかし、ワークフローは業務の流れを支援し、自動化するもので、グループウェアとは本来の目的が異なる。

3 「情報の共有」を可能にする経営情報システム構築技術の発展

(1) LAN 活用技術の進化

従来の汎用コンピュータの世界では、システムのハード構成は、極端にいうとホスト機と端末しかなかった。アプリケーションの開発も、COBOLで組むしかなかった。あるいは、数種類の第4世代言語（4GL）を選択するしか選択肢はなかった。要求分析の技術、プログラミングの技術が強く要求され、仕様書の作成とそのコーディング（つまりプログラム作り）の分業体制でシステムの構築をした。

しかし、エンド・ユーザー・コンピューティングを可能にするオープンシステム化の時代を迎えて、アプリケーションを開発する環境と、これをささえるアーキテクチャーは大きく変わりつつある。プログラミング技術は、流通ソフトを組合せてシステムを構築する「組合せ技術」に変革した。ノンプログラミング言語による流通ソフトの組合せでシステムを構築する方向に変わりつつある。

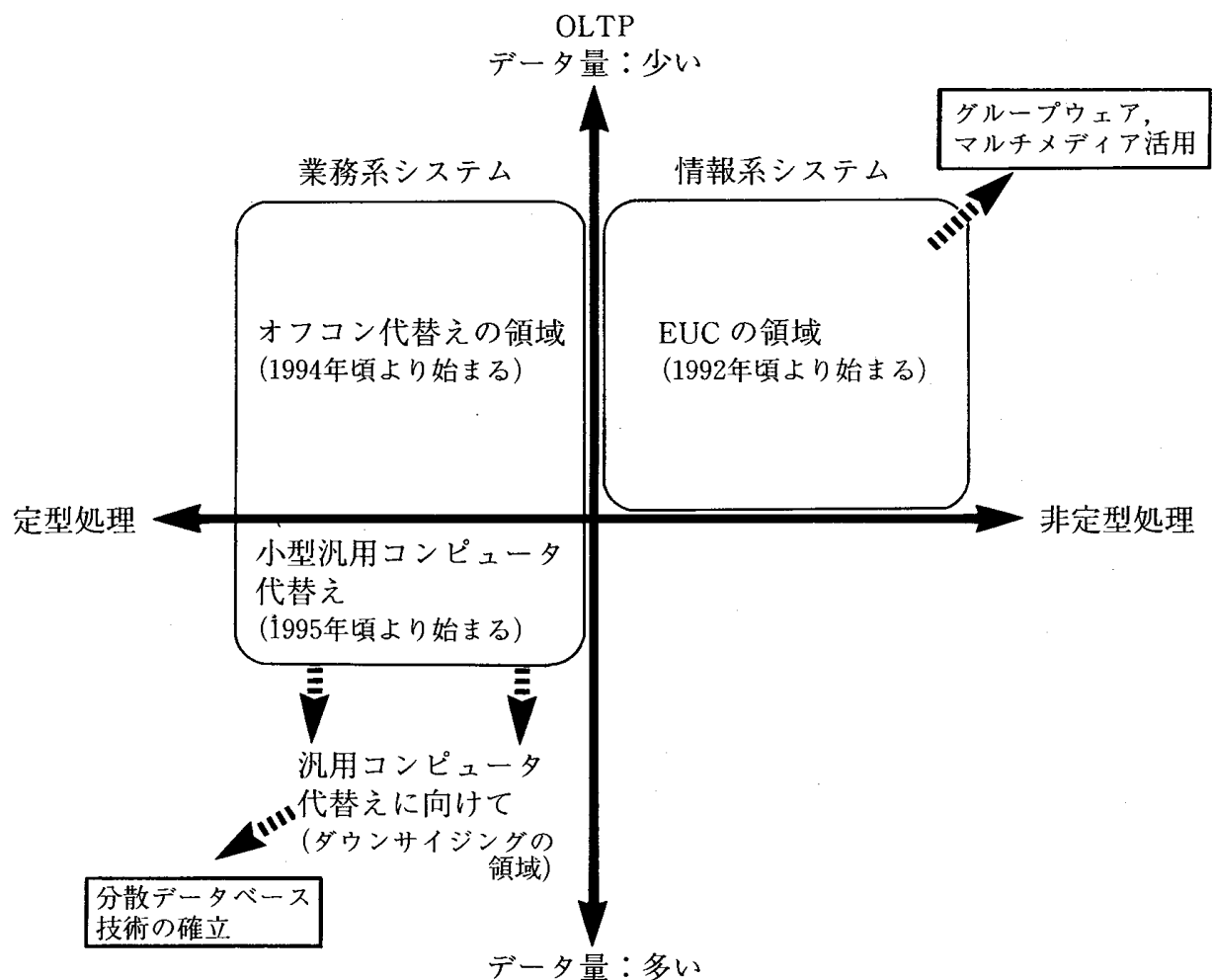
しかし、コンピュータに関するかぎり前提条件を明確にして説明しないと誤った解釈がなされることが多い。そのため、業務系システムと情報系システムを定義する。

業務系システムというのは、受発注オンライン・システム、在庫管理システム、給与計算システムなどで、企業や部門の事務処理を合理化するためのシステムである。基本的には定型処理である。

情報系システムというのは、業務系システムが蓄積した情報や外部データベースの情報を、ユーザー自身で自由に検索・加工し、経営計画の立案や、管理のための意思決定に役立てるためのシステムである。これは基本的には、非定型処理であり、これまでの汎用コンピュータが苦手とする領域である。

図2に「クライアント・サーバー・システムの活用領域の進化」⁵⁾を示す。

図2 C/Sシステムの活用領域の進化



「エンド・ユーザー・コンピューティング (EUC) の領域」に示すように、これまでの汎用コンピュータの苦手の、この領域の情報系システムからパソコン LAN の活用が始まった。それは1992年頃からで、それが次第に業務系システムであるオフコンの代替え領域である定型処理のデータ量の少ない領域に広がり、さらに小型汎用コンピュータで処理したデータ量の多い領域に拡大を続けている。今後は、汎用コンピュータ不要の時代がくることであろう。一般にパソコン LAN の解説書が多く出ているが、このように処理形態毎にシステム開発手法が異なるという説明が少ないので、このことに焦点をあてて解説する。

(2) Windows 環境の出現

a. ウィンドウ環境とは？

ウィンドウ・システムとは、あたかも机の上にいろいろの書類を広げるようにコンピュータの一画面上にいろいろの書類の窓をあけ、これら一つ一つの窓のなかでそれぞれ異なる処理が同時実行（マルチタスクという）することができる。たとえば、データベースの蓄積データを呼びだしながら、表計算ソフトを利用し、それに呼びだしデータを貼り付けたり、完成した表をグラフにして、経営者に分かりやすくして提出する。さらに、そのグラフをワープロで作成した文章のなかに挿入するというようなことが可能になる。この機能によりユーザー・インターフェイス（操作性）が格段に向上する。

これまでのウィンドウ・システムは UNIX 環境の X-Windows がよく知られていた。ウィンドウ・システムは、オープンシステムの構築や、クライアント・サーバー・システムにおいて極めて有効であることが、UNIX 環境で実証されていた。

さらに、これに不可欠な GUI (Graphical User Interface の略) と API (アプリケーション・プログラムとシステム：OS とのインターフェイス) について理解しなければならない。GUI はユーザー・インターフェイスを高めるために開発されたもので、ウィンドウ・システムの画面上にさまざまなグラフィック (アイコンという) を表示するものである。ユーザーはコンピュー

タの画面指示にしたがって、マウスを操作してアイコンに指示を与えるだけでコンピュータに命令を与えることができる。

API は開発するアプリケーション・プログラムとシステムとのインターフェイスを規定することによって、その移植性を保証しようとするものである。これにより、ユーザーは、購入したソフトウェアをいろいろなパソコンで利用することができ、マルチベンダー環境が出現する。これにより、ソフト開発企業においても、自社の製品があらゆるハード・ベンダーのパソコン上で作動することになる。

これはオープンシステムの構築にとり極めて重要なことであるが、共通のアプリケーション・プログラムを作成するためには、プログラム言語が規定されているだけでは十分ではなく、OS（基本ソフト）や、周辺機器へのアクセス、GUI やウインドウの制御、他のシステムとの通信の方法など多くの規定が必要となる。

b. 「Windows3.1」の出現

マイクロソフト社が前述のようなオープンで使いやすいウインドウ環境を実現するために、1992年5月「Windows3.1」の日本語版を発売した。これは、これまでの MS-DOS に比べて、次の特徴がある。

- GUI と操作性の統一：ソフトの基本的な操作を共通化して GUI と呼ばれるアイコンやボタンをマウスで操作することにより、誰でも使える易しさを目指した。
- オープン化：特定のパソコンや周辺機器に依存するのではなく、どんなハードやソフトでも使えて、パソコンの活用範囲を広げる。
- OLE 機能：複数のソフトが同時に使えるように Windows の特徴をだすために、ソフト間で直接、即時的にデータを連携させて使える OLE（オーレという、後述）という機能を内蔵している。これにより、複数のソフトをデータ連携で結び付けることができる。
- マルチタスクが可能：複数のソフトを並列に実行するマルチタスクが可能。例えば、データベースの蓄積データを検索しながらパソコン通信を

することが可能。

- ・マルチメディア対応：WYSWYG（画面で見たとうりの印刷結果をうる
こと）等の機能が充実。

その後、マイクロソフト社は、これよりさらに充実した「Windows95」
の日本版を1995年11月発表した。

c. イベント駆動型のプログラミング

Windows には、イベント駆動型のプログラミングという手法がとられて
いる。これは MS-DOS 用の手続き型のプログラムとは著しく異なる。これは
GUI を駆使した、開発スタイルで、プログラム構造を設計し、ソース・コード
をコーティングするという、これまでのプログラミングと対立する方法である。

これまでのプログラミングのスタイルは、処理 1，処理 2，処理 3…とい
うふうに記述した。これはプログラマーが処理の順番を決めることができ
ることを意味する。

Windows 環境のプログラミングは、この考え方を頭から消すことが求め
られる。プログラムの制御の主導権は、Windows 本体の方にあり、作成す
るプログラムは、Windows 本体から順不合で渡されるメッセージに正しく
対応するように作らなければならない、という点が著しく異なる。つまり、
Windows 環境では「処理 1 が行うべきメッセージが来た後に、処理 2 を行
うメッセージが Windows 本体から来る」ことはまったく保証されていない。
そのため、各メッセージに対する処理を並列的に記述する。

このため、Windows プログラムは大きく分けて、「Windows からメッセ
ージを受けて処理を依頼する制御部」と「受けたメッセージに対する処理を行
うウインドウ・プロシジャ」から構成される。

この考え方は、オブジェクト指向のプログラミングであり、イベント・ド
リブン（イベント駆動型あるいは、メッセージ駆動型）といわれるプログラ
ミングである。

DBMS（データベース管理システム）を核にするクライアント・サーバー・
システムの構築手法が固まってきたことを背景に、画面設計（GUI 構築）

だけでなく、データ管理 (DBMS の操作) までも GUI 上で行うことのできるイベント・駆動型のビジュアル開発ツールが登場してきた。ビジュアル開発のメリットは、エンド・ユーザー・コンピューティングのための開発期間の著しい短縮に貢献している。

d. イベント・ドリブンの Visual Basic

Unix の X-Windows や、Macintosh, Windows などの GUI 環境の必須プログラミング・スタイルが、イベント・ドリブン (駆動) である。

ノンプログラミングのビジュアル開発ツールとして脚光を浴びている Visual Basic は意味は “Visual” な “Basic” となるが、これは従来の Basic 言語ではない。イベント駆動型の開発ツールである。

Visual Basic は、左側に21個のボタンからなるウインドウはツール・ボックスと呼ばれている。そのアイコンによる矢印 (↑) という表示以外は、全部すべてコントロールである。Visual Basic のプログラミングは、マウスによってコントロールを、フォームと呼ばれる作成するアプリケーションの入出力インターフェイス (画面) に配置することによってなされる。マウスのクリックやドラッグなどのイベントの発生に応答して、コードが実行されるので、イベント駆動型のプログラム言語であり、極めて使いやすい。従来の Basic 言語のプログラムのように、ユーザーの操作手順によって時系列的にコーディングする必要はない。ユーザーのマウスの操作がこれを決める。(これを external control と呼ぶ) これは Lotus 123等の作表ソフトで作表した表をグラフ化するとき、マウス操作だけで直ちにグラフ画面ができるのと同じイメージである。

e. カスタム・コントロール：VBX

前項で Visual Basic のコントロールという概念を説明した。コントロールを組合せることによって、プログラムの作成が可能になる。一般に、Visual Basic に標準で添付されているコントロールを標準コントロールという。これにたいして、サード・パーティにより提供されるコントロールをカスタム・コントロールまたは VBX という。ファイル名の拡張子が VBX である

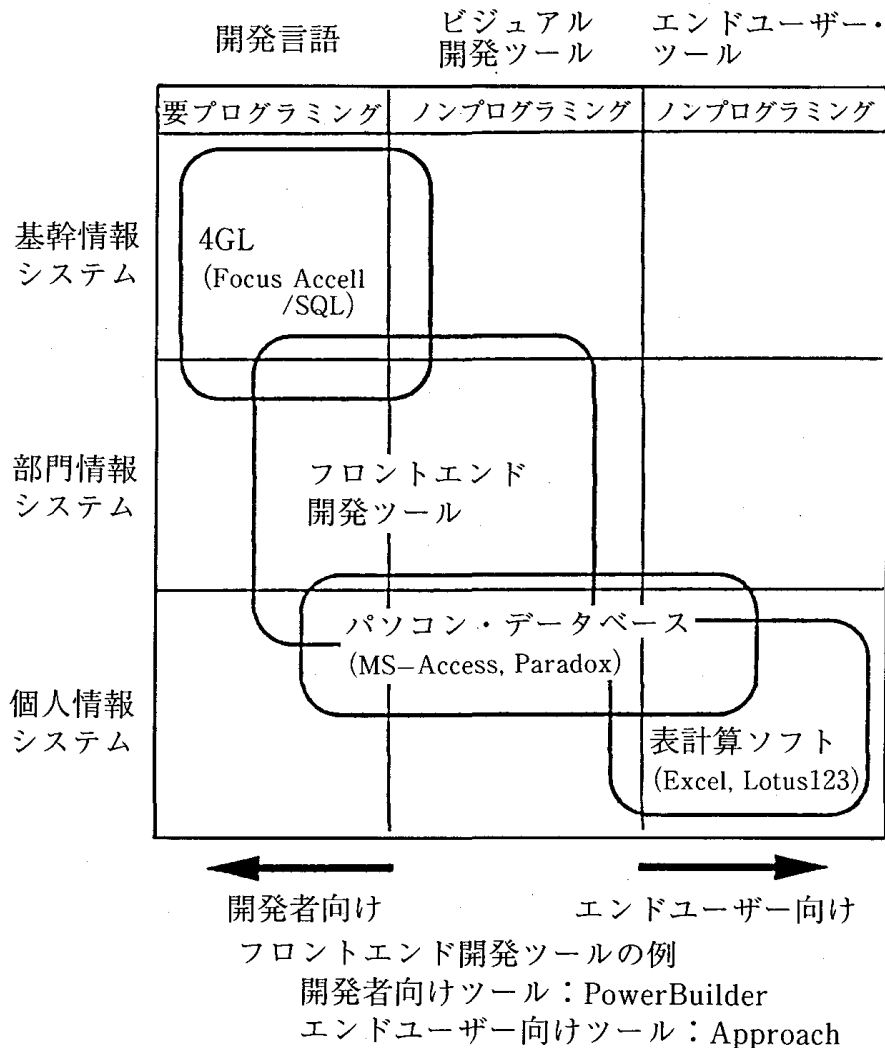
ことからこのように呼ばれる。

これには2種類ある。標準コントロールを拡張した代替えコントロールと、もう一つは、標準コントロールとはまったく異なる機能を提供するコントロールである。米国では2百社以上のVBXを提供する企業があり、多くのVBXが流通している。その代表的なコントロールは、メインフレーム環境との通信、SQLデータベースへのアクセス、パソコン通信、3次元ブラフの作成、アニメーション作成等である。

このようにして、ソフトの部品化を促進して、プログラミングは、ソフト部品の組せに移行したことに注目していただきたい。これは将に革命的な開発環境の変革である。

(3) ビジュアル開発環境の位置づけ

図3 ビジュアル開発ツールの位置づけ



備考：図3に示すフロントエンド開発ツールは、開発者向けツールとエンドユーザー向けツールは同じWindows上で開発するために、両者の距離は次第に接近することであろう。将来は、ソフト産業は、このようなノンプログラミング・ツールを作るコンポーネント・ビルダーとそのコンポーネントを使ってシステムを構築するシステム・ビルダーに2極化することであろう。

前項までのVBXの登場で流通ソフト部品を組合せるだけでアプリケーションの開発が可能になることが理解できたであろう。これまでの汎用コンピュータ時代のソフト開発は、一つの大きな固まりとしてのプログラムという製品をSEとプログラマーによって人海戦術的に作りあげるというきわめて労働集約的な産業であった。自動車の製造は、ビスやナットなどの部品やタイヤ、エンジン等を外注して、製品を組み立てるのと同様に、ソフト部品の組合せによって、ソフト製品をつくりあげる時代が到来したのである。

図3に「ビジュアル開発ツールの位置づけ」²⁾を示すが、図2と対照して理解して頂きたい。その範囲は次第に拡大して、恐らく数年を経ずして、すべての業務が、流通ソフトの組合せだけで構築可能な時代が来ることであろう。

流通ソフトの組合せでシステム構築の時代が来ると表現したが、工場で、部品の組立をロボットにやらせたり、ファクトリ・オートメーションにより自動化できることに注目していただきたい。これからのソフト開発は、流通ソフトを組合せることと、ソフト間の連携技術が重要となる。工場で部品を取付けるときに部品と取付け対象間の整合性の確保が必要である。これと同じことを可能にするのが、OLEという考え方で、これを自動化するOLEオートメーションという考え方も登場したが、これについては後述する。

用語解説

DBMS: データベースを管理するシステムである。ファイルはデータの記憶が可能であるが、その記憶情報を検索する場合には、その記憶番地を指定しないと検索ができない。このことは、検索プログラムとデータが一体になっていることを意味する。これに比べて、データベースは、どのプログラムからでも検索できるように検索プログラムから独立になっている。そのかわり、データの倉庫であるデータベースには、DBMSという倉庫番がいて、この倉庫番がデータの所在番地を覚えている

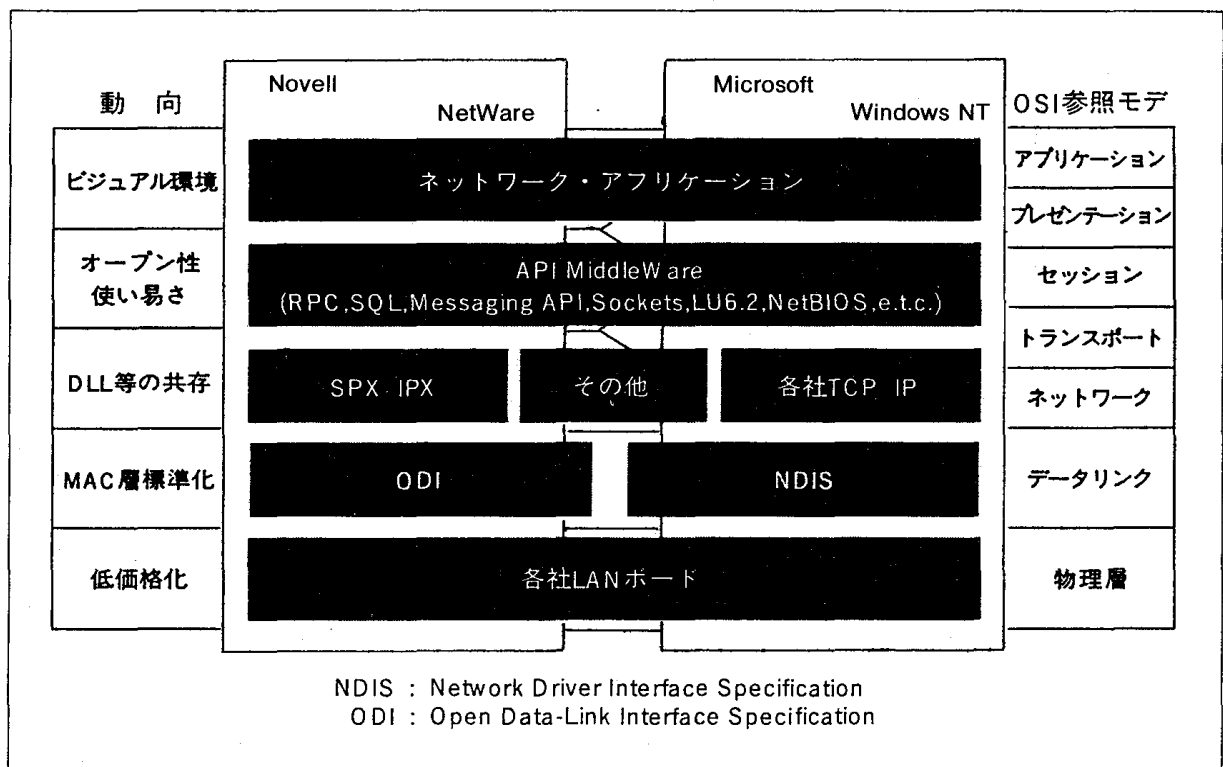
ので、DBMS に指示するだけで、そのデータを検索してくれる。

(4) デファクト・スタンダード化する LAN プラットホーム

デファクト・スタンダード (De-fact Standard) とは、世の中に普及して結果的な標準製品となって独占ともいえる圧倒的な強さを誇る製品のことである。かつての NEC の PC98 や最近のマイクロソフト社の Windows などはその例である。

図 4 に最近のパソコン LAN プラットホームのデファクト・スタンダード化する動向を示す。

図 4 パソコン LAN プラットフォームの動向
(「ミドルウェアを制すれば C/S 構築は“百戦危うからず”」ネットワーク・コンピューティング誌 (1995年 6 月号, 34 頁・図 3 を転載)



左にネットワークの分野で参照される OSI モデルを、右に最近の動向を示す。パソコンに OS (基本ソフト) があるごとく LAN にもネットワーク OS (基本ソフト) が必要である。

ネットワーク OS として、ノーベル社の NetWare が広く普及していたが、

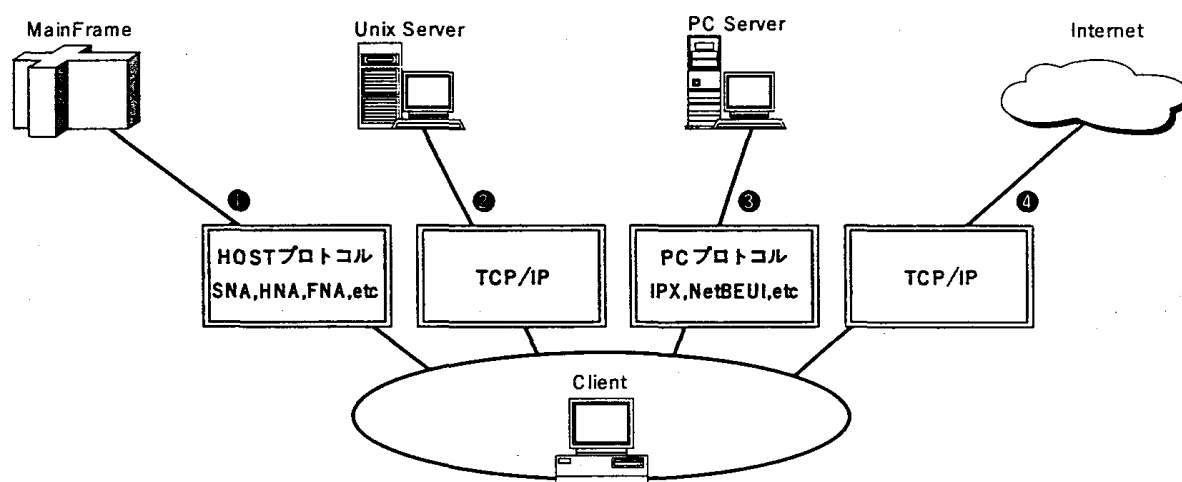
これからはマイクロソフト社の Windows NT が普及することが期待される。このためこの2つを分類して示している。

下層レイヤから順を追って述べると、かつては20万円もした LAN ボード (NIC: Network Interface Card) が、現在では2万円以下で入手できる時代になった。この製品選択基準は廉価で ODI や NDIS という標準インターフェイスを添付していることを確かめるだけでよい。

上位層もネットワーク OS が普及したことにより、ファイル (データベース) やプリンターの共有を目的とした部門 LAN クラスになれば、SPX/IPX プロトコルとの共存を考慮した相性のいい TCP/IP ソフトを選ぶだけでよいという状況となった。図5に閉鎖的な各社のメインフレーム・プロトコルから、オープンなプロトコルに転換した UNIX サーバー, PC サーバー, インターネットとの通信プロトコルを示す。

ここで LAN 特有の API: ミドルウェア層の製品選択の問題に突き当たる。この層の製品選択の幅は多数で、この選択のポイントは、「使いやすくてオープンな環境の実現」と「GUI ベースのビジュアルなアプリケーション」である。このことは後述する。

図5 閉鎖的からオープン・プロトコルへの転換



用語解説

・OSI 参照モデル

ネットワークはパソコン間を回線で接続しただけではデータ通信はできない。これは物流層で、そのうえにデータを流す方法を規定するデータリンク層、交換接続する方法を規定するネットワーク層等7層のプロトコル（通信手順）を規定しそれを送受信するパソコン間で遵守することによってオープンなネットワークを構築することができる。この取り決めの参照標準モデルのことである。

備考：ネットワーク OS

ここでは、Windows NT と NetWare しか述べていないが、1994年末から95年にかけて、パソコン用のネットワーク OS が相次いでバージョン・アップをした。94年11月日本 IBM が「OS/2サーバー J 4.0」を発表した。同12月、マイクロソフト社が Windows NT 3.5」を、さらにノーベル社が「NetWare 4.1J」を発表した。

これらのネットワーク OS は、当初のファイル／プリント・サーバーとしての機能だけでなく、大規模システムへの対応を想定した機能を盛り込んできた。すなわち(1)複数サーバーを一元管理する機能、(2)マルチベンダー環境への対応、(3)マルチプロセッサ対応によるスケーラビリティの拡張、といった機能である。

備考：図5の解説追加

ホスト接続用のソフトウェアもネットワーク OS と無関係ではない。LAN 経由でホストと接続するケースが増えているからである。LAN とホスト・コンピュータを接続するには、ゲートウェイとなるハードウェアあるいは、ゲートウェイ・ソフトを導入し、端末のパソコンに対応する端末エミュレータを使わなければならなかった。しかし、Windows NT では、マイクロソフト社が発売した、「Microsoft SAN Server」がゲートウェイの標準になりつつある。その理由は、SAN Server に対応した、端末エミュレータが、サード・パーティから相次いで登場しているからである。

さらに、Windows NT では TCP/IP, IPX/SPX, NetBUEI, Apple Talk など、現在の LAN で使われている主要通信プロトコルを標準でサポートしている。さらに、NetWare や UNIX などの環境と共存するための機能も盛り込んでいる。UNIX の世界標準の TCP/IP に関しては、すべてのネットワーク OS がサポートしている。

4 これまでのコンピュータとパソコン LAN との構築手法の変革

(1) ソフトウェアは作らない、流通ソフトを組合せる

クライアント・サーバー・システム（C/S システム）が全盛期を迎えつつある一方で、一時代前とは異なった意味での技術者不足が叫ばれている。従来の汎用コンピュータや、オフコンとは全く異なる構築技術が求められるか

らである。

C/S システム構築の目的は、大きく分けて次の3つに分類される。

- 1) 従来の汎用コンピュータやオフコンでは苦手であった非定型処理をパソコン活用のエンド・ユーザーにやらせて、情報の創造・蓄積をして、経営を活性化する。
- 2) ビジネス・プロセス・リエンジニアリング (BPR と略称) に代表されるように顧客に便利な経営システムに再構築して、全く新しいパラダイムのシステムを構築して経営の生産性を向上させる。
- 3) システムをできるだけ早く安価に構築する。

この第3項に関しては、コボル比10倍以上の生産性の向上が可能で、しかもエンド・ユーザーの要望を満たすに最適なシステムの構築が可能となる。

その理由は、ソフトウェア構成は、できる限り市販のパッケージ・ソフト（これは流通ソフト、または流通モジュールということにする）組合せることで可能となる。これからの経営情報システムを構築する技術者には、この流通ソフトをどのように組み合わせたら最適なソリューションが得られるか、このような総合力と創造力が求められる。

したがって、これまでのメインフレームやオフコンと全く異なる開発手法が求められる。

- (2) 設計とソフト開発は分離しない（仕様書なしの開発も可能）

C/S システムはエンド・ユーザーの満足度の高いシステムを早期に開発することができる。これまでの開発は、システム設計とソフト開発を分離して、分業体制でシステムを構築していた。しかし、C/S システムではシステム設計と開発は分離しないで、GUI（パソコン画面）上で、エンド・ユーザーの要望を確かめながらシステムの構築が可能である。

つまり、パソコン LAN のシステム構築は、これまでのように専門化した開発チームが順次自分の分担を果たしていく形態ではなく、一人あるいは、一つのチームが一貫してプロセス全体を実現し、ユーザーに対して早期にシステムを提供することが可能となる。

しかも設計の視点は、これまでのようにファイル中心に一貫したシステム構築ではなく、イベント駆動型のプログラミング（前述の 3, (2), c 参照）で開発ソフトをモジュールに分割して多用することができ、この面でも飛躍的にソフト開発の生産性を高めることができる。

このパソコン構築手法を実現しこれを経営手法に応用したために、前述したアメリカの企業が驚異的な業績回復を成し遂げたのである。BPR やコンカレント・エンジニアリングという新しい経営手法はこの方式を採用している。

(3) パソコン LAN 開発に必要な技術能力

これまでの汎用コンピュータやオフコンは、端末とコンピュータが一体で、詳細なハードの知識がなくてもコボル言語さえ修得すればシステム構築が可能であった。C/S システムでは、クライアントとサーバー間にデータをやり取りするミドルウェア・ソフトという問題がでてきた。しかも、採用するパソコン、サーバー等のハードの知識から、ネットワーク OS、開発支援ツール等の幅広い選択肢のなかからどれを選ぶか、という知識がないと、流通ソフトの選択・組合せができない。

パソコン LAN はネットワークで、データの流れを水の流れに例えると、パイプの細いところで、水の流れが制限される。これをボトルネックの発生という。このボトルネックが発生するとシステム全体がダウンする。時々、郵便貯金や、銀行の現金支払機が連休前の繁忙期等にダウンするのはすべてこのボトルネックの発生が原因である。

LAN や LAN-WAN-LAN 等の広域 LAN 或いは大規模 LAN になると、このボトルネックの回避が LAN 構築にとり非常に重大となる。広域 LAN になると WAN の部分が公衆通信回線や伝送容量の低い専用線を活用するので、ここにボトルネックが発生することが多い。

このためネットワークに流れるトラフィック（データの量）を測定して、何処にボトルネックが発生するかを事前に予測して対策を立てることが求められる。

このため、これからの C/S システム構築の技術者には、流通ソフトの組合せ技術とハードウェア、特にネットワークの幅広い知識が必要となる。

これまでの業務の要求分析とその仕様書を作成してコーディングする、という汎用コンピュータ時代のシステム構築とこれからの C/S システム時代のシステム構築の大きな違いを認識していただきたい。

(4) スパイラル型の三段階のシステム構築

エンド・ユーザー・コンピューティング時代のパソコン LAN の構築は、これまでのようなシステム設計と開発の分離ではなく、図 6 に示すシステム設計と開発一体型の「三段階のスパイラル（うずまき）型のシステム開発」手法を提案する。

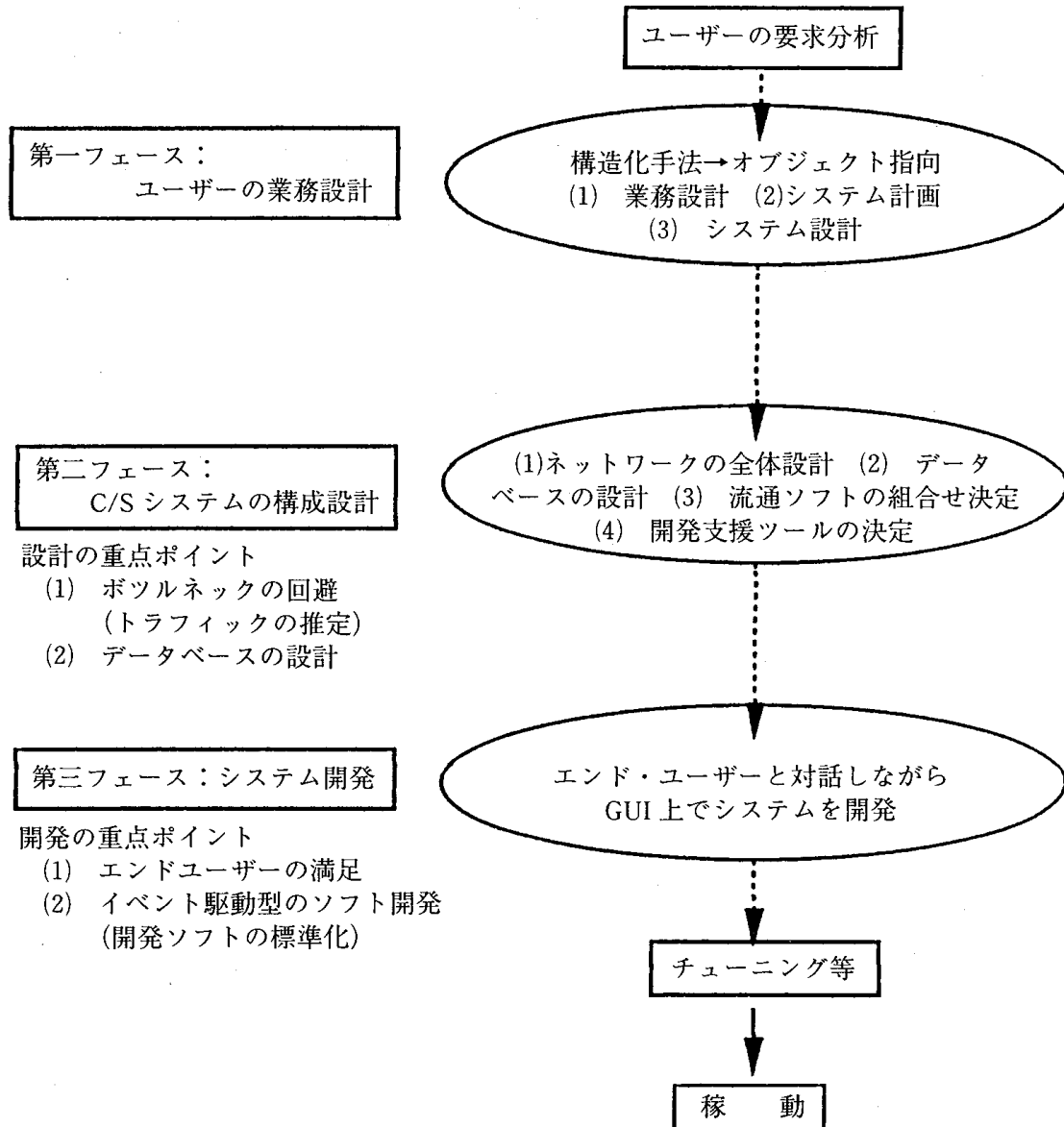
第一フェーズは、ユーザーの要求から分析が始まり（これまでの構造化手法からオブジェクト指向に変革した）第 2 フェーズは、1) ネットワークや開発支援ツールの決定等で、スパイラル的に相互に関連さしながら決めていく。特に、設計の重点ポイントとして、ネットワークに流れるデータ量（トラフィック）を推定して、ボトルネックの発生しないような設計が必要である。さらに重要なことは、データベースの設計で、リレーショナル・データベースを採用することが多いので、そのアクセス時間を短くするために、データの正規化、非正規化をどうするか、というノウハウの蓄積が必要である。C/S システム構築において、このデータベースの構築が成功すれば、あとは組合せソフトの選択を誤らなければ、成功間違いない。

第三フェーズは、「エンドユーザーと GUI（画面）上で対話をしながらシステムを開発」する段階である。これによりユーザーの満足度の高いシステムの構築ができる。ここでイベント・ドリブン（駆動型）のシステム開発のコンセプトを生かして、ソフト・モジュール（部品化）の標準化を推進することが必要である。

これまでのシステム開発は、完成しないとユーザーの意見を聞くことができなかったのに比べて、格段の進歩であり、ユーザーの満足度の高いシステムの構築ができる。勿論、データベースの設計が成功しておれば、エンド・

ユーザー自身のシステム開発も十分可能である。

図6 三段階のスパイラル型システム開発



5 クライアント・サーバー・システムの構成要素の進化

(1) 分散オブジェクト指向に進化する C/S システム

C/S システム（クライアント・サーバー・システム）で普及しているのは、UNIX 系サーバー（リレーショナル・データベース・マネージメント・システム：RDBMS と略称）と Windows 系のパソコンをクライアントとしたシ

システムである。RDBMS に「Oracle」を採用し、クライアントのアプリケーションはマイクロソフト社の「Visual Basic」を使って開発するのが多い。このようなシステムを構築する場合にミドルウェアに何を使うのか決定が一番難しい。

『ネットワーク・コンピューティング』（株・リックテレコム刊）1995年6月号、「一から学ぶ C/S のソフトウェア構成術」³⁾（30－33頁）を参考にその技術進展の状況を解説する。

図7に示すように C/S システムは当初（1992年頃）は単純二階層型システムとして、パソコン端末からサーバーのデータを活用して非定型処理をするところから始まった。

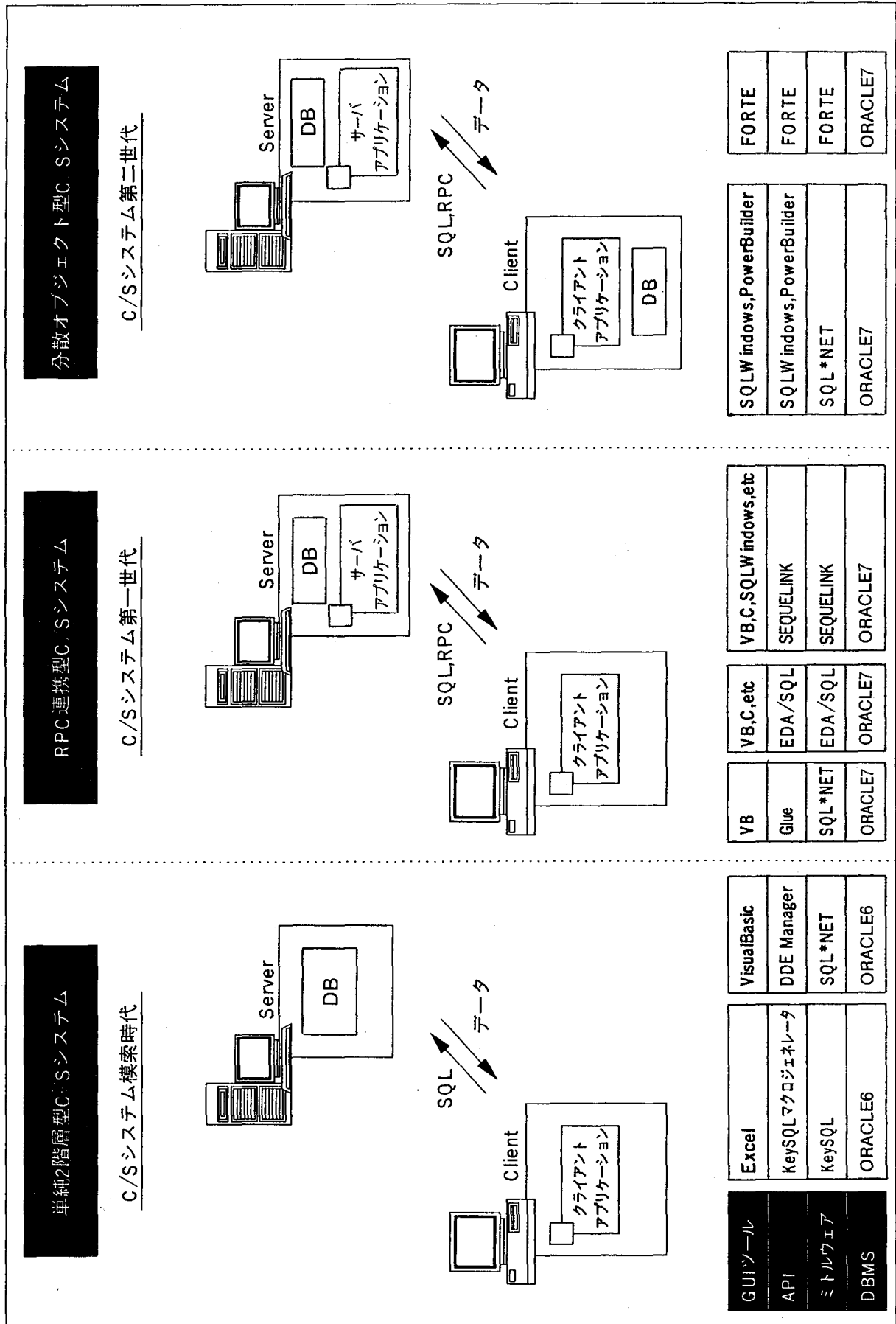
その後、OLTP（オンライン・トランザクション処理）の増大によるボトルネックの発生を除去するために、現在、多用されている RPC 連携型の第一世代へと進化した。これが、第二世代では分散オブジェクト型の C/S システムに進化することを図は示している。「この世代交替を可能にするものは、サーバー側の RDBMS とクライアント側のアプリケーション開発ツール、そしてクライアント・サーバーの両方にまたがるミドルウェアと API の進化である。」

単純二階層型のシステムでは、基本的にアプリケーションをクライアント側に配置して、データベースに対する入・出力だけをサーバー側に処理させる方式である。クライアント側からデータベース問い合わせ言語である SQL 命令文をサーバーに発信し、サーバーからはその処理結果を受け取るという単純な構成である。パソコンを活用して非定型処理するには向いていた。

しかし、このような構成では、ネットワークを流れるトラフィック（データ量）が多くなり、クライアントの数の増大に応じてボトルネックが生じ、システムダウンが多くなる。またクライアント側の負荷が重く、同一の処理を繰り返し集中する OLTP には不向きである。

そこでこれを改善した、第一世代の RPC 連携型システムが登場すること

図7 クライアント／サーバー・システムの世代変化



になる。

RPC (Remote Procedure Call: 遠隔手続き呼出し) とは、サーバー側にもアプリケーションをおき、クライアントと処理を分散する方法が考えられた。

RPC はRDBMS に組み込んだ処理ロジック (手続き) を呼び出すことで、クライアントとサーバー間の処理の連携が行われる。「一連の定型的なデータベース処理をサーバーに集中し、クライアント側は、画面や入出力のユーザー・インターフェイスの制御を受け持つことになる。」

RPC 連携型システムは、運用段階を見越した設計が必要で、どうしてもシステム設計が複雑になるという欠点があった。これを解決するために95年代に登場したのが、第二世代のオブジェクト指向型の C/S システムである。このことは後述する

(2) ミドルウェアの役割

Windows アプリケーションと RDBMS も接続するのがミドルウェアである。これには2つの側面がある。パソコン・ユーザーが表計算ソフト (Excel 等) を活用している場合に、データベースの検索機能を提供するツール (これをクエリー・ビルダーという)、つまりフロント・エンド・ツールとしての側面である。もう一つは、いろいろなマルチベンダーのデータベースを活用する場合にマルチ・データベースに接続する「コネクティビティ・ツール」としての側面がある。

ソフトウェアのタイプには RDBMS のように明確な論理に基づいて製品化したものから、マーケットにおける個々の製品化が先行して定義づけのあいまいな製品まである。ミドルウェアは将に後者の製品群である。その共通機能 (サーバーへのアクセス) から考えると次の5つに分類することができる。

1) エミュレーション型

これは図5で「閉鎖型からオープン・プロトコルへの転換」を示したが、図示の(1)、(2)に示すメイン・フレームや Unix サーバーへアクセスしてマイクロ・メインフレーム・リンクを実現するためのツールで、3270端末エミュレーションや VT 端末エミュレーションに相当する機能を提供するものであ

る。

2) アプリケーション会話型

これは図5の①に示すように、クライアントとメイン・フレームのアプリケーションが会話をするタイプで、IBM・SNAのLU6.2プロトコルとクライアントのAPPCの組合せが、ピア・ツ・ピア通信を可能にする。

3) リモートDBアクセス型

図5の②に示すようにサーバーのデータベースを検索するもので代表的なミドルウェアの機能である。

4) リモート・プロシージャ・コール型

これも図5の③に示すリモートDBアクセス型であるが、アクセス時間を短縮したり、ボトルネックを回避するために、リモート・プロシージャ・コールを付けたミドルウェアである。

5) ファイルアクセス型

これは図5の②、③、④のファイルをアクセスしたり、ファイルを転送するための機能を提供するミドルウェアである。製品としては、TCP/IPのFTやNetWare製品群がある。これはマルチベンダーのデータベースをアクセスする機能を有するミドルウェアである。最近ではInternetで採用されるMosaic/WWWなどがある。

これらの製品群は、現在では半年毎に製品が入れ替わるぐらいの激しい競争を繰り広げている。そのうちにミドルウェアもデファクト・スタンダード(業界標準)化してLANが構築しやすくなることであろう。

(3) データベース技術の発展

a. データ中心のシステム設計

これまでのシステム設計は、プロセス中心のシステム設計であった。ある大手のシステム部長は、「データは骨格、プログラムは肉。骨格を整えないで肉付けだけをして意味がない。データ項目がどう定義され、システムにどう使われているか、逐一現場で知り管理していなければアプリケーションはEDPS部門依存型になってしまう」といったのは、1982年のことである。

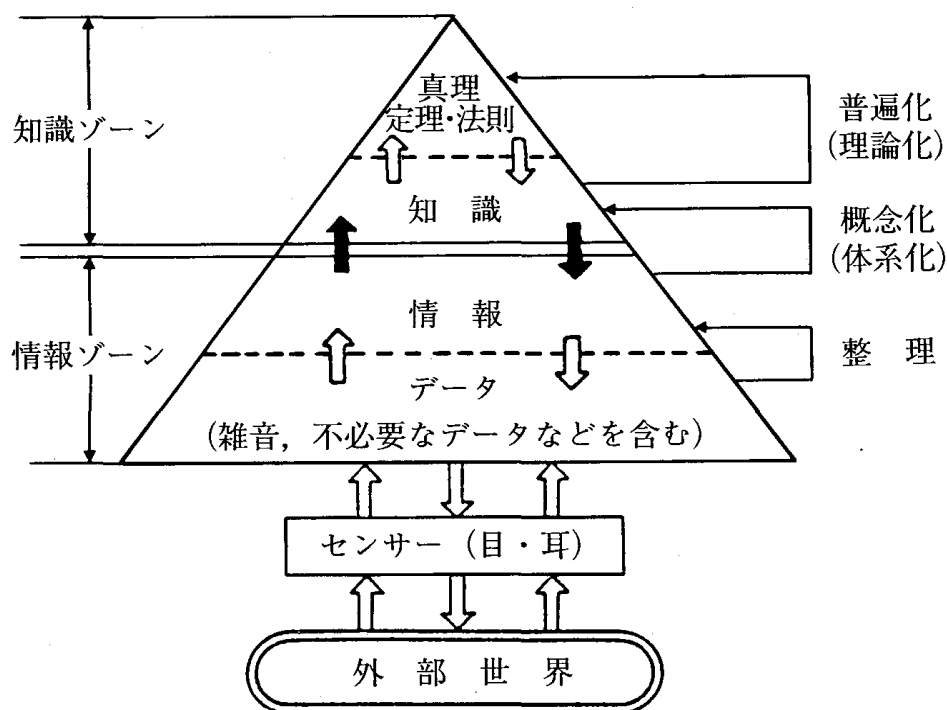
冒頭で述べたように、これからの経営情報システムは「情報の共有」をキーワードにして構築すべきことを述べた。

特にC/Sシステムの構築において、データベースの構築に成功すれば、システム構築の大部分は完了したようなものである。これからの情報の共有を目的とするシステム構築において、データベースは企業の最大の経営資産である。

システムを再構築する場合に求められるのは、業務システムの見直しではなく、企業の資産としての情報資源の基盤整備である。これはこれまでのプロセス（処理）中心のシステム構築を見直して、データ中心のシステム設計にやり直すことを意味する。

では、このシステム再構築の出発点として何から始めればよいのだろうか。まず、コンピュータの入力原始メモ（伝票）に使う用語を統一して、用語定義書を作成し、これを各部署に回して全社的に同じ用語を利用するようにすることである。

図8 データ、情報、知識の違い



(出典) 和多田作一郎「AIの基礎を知る事典」、実務教育出版を加筆修正

たとえば、売上という科目一つを考えても、製品売上もあるし、工事・サービスの売上もあり、外注先への部品の有償支給もある。これらをそれぞれ明確にしないと原価計算があいまいとなり、売上の分析比較の基準が狂うことになる。

これからの経営情報システムの目的は、経営の実体の把握と経営活性化情報の創造と蓄積・活用であり、いずれも経営戦略を支援するシステム構築にとって不可欠のものである。

このため、図8に示すように「データ、情報、知識の違い」を認識して、データ中心のアプローチ（DOA という）を行うことが必要である。

b. ファイルとデータベースの違い（DBMS の登場）

図9にファイルとデータベースの違いを示す。図の(a)に示すようにデータは階層構造をしていて、例えば衣料品売場（大分類）は婦人服、紳士服、子供服といった部門（中分類）のファイルから構成される。この大分類が、商品情報データベースに記憶されているとする。このファイルは、スカート、セーター、ブラウスといった品種（小分類）によって構成され、これをレコードと呼ぶ。さらにレコードは、色別、型別、サイズ別などの品目（単品）から構成され、これをフィールドという。

このデータベース、ファイル、レコード、フィールドはコンピュータ用語で、このファイルとデータベースは図の(b)に示すように根本的な違いがある。ファイルは、(b)のように処理プログラムと一体となって構成されていて、記憶商品を検索する場合には、ファイルの何処にある商品かということのを的確に指示することが必要である。

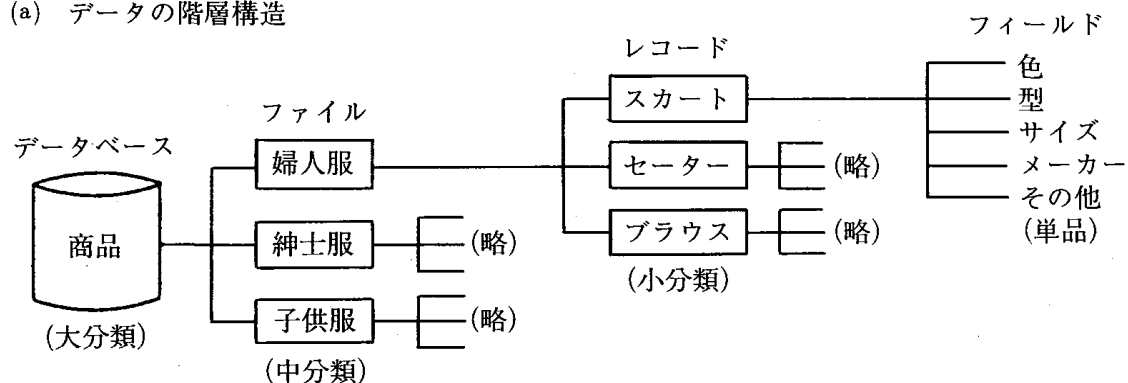
ところが、データベースはデータを蓄えるベース（倉庫）という意味で、ここにDBMS（データ・ベース・マネージメント・システムの略）と称する倉庫番のようなプログラムがあり、この倉庫番がそのデータの所在を知っているので倉庫番に必要なデータを指示することによって、それを探してくれる、という特徴がある。

この RDBMS として現在 C/S システムに使われているのがリレーショナル

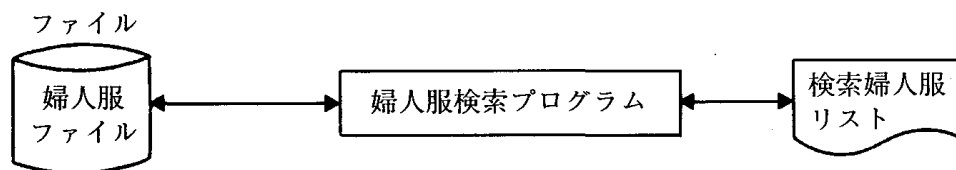
ル・データベース（RDBMS）である。これが今後、オブジェクト指向データベースに移行する傾向が1995年頃からでてきた。

図9 ファイルとデータベースの違い

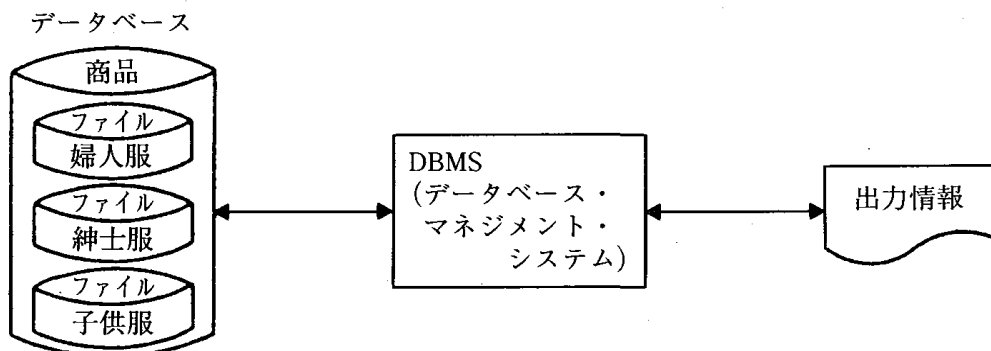
(a) データの階層構造



(b) ファイルとプログラムの関係



(c) データベースと DBMS



c. リレーショナル・データベース

C/S システムの構築において、「Access」や「Visual Basic」（以下 VB と略称）などをフロントエンドにした業務システムの開発が盛んである。使いやすく美しいユーザー・インターフェースを持つパソコン・クライアントと強力な堅固なデータベース・サーバーの組合せが魅力的である。しかし、

一方で、思ったほどの性能がでない、保守が大変だ、新規アプリケーションの開発に時間がかかる、という悩みを持つ企業が多い。

この悩みの生じる原因は、大部分、リレーショナル・データベースの実務知識を有する技術者の不足による。LAN と RDBMS の組合せでは、設計の良否が完成システムの良否に大きく影響する。実際、開発の遅延や失敗は、新しい開発支援ツールの使い方に関するノウハウの不足と並んで、データベースやアプリケーション設計の不具合が原因となっていることが多い。

Access や VB の登場で、RDBMS の活用は非常に簡単になったかのようにみえるが、それらは決して処理を自動化してくれるわけではない。RDBMS の理論上の前提や、現実の抱える実装上の理論的な前提や、現実の製品の抱える実装上の制約を無視すると、思わぬ失敗をしてしまう。プログラミングが楽になった分だけ、データベースの設計の重要性が増したのである。

図10に「ファイルとリレーシヨンの概念」を示す。ファイルというのは、(a)に示す表のことで、図9の(a)のデータが記憶される。縦一列のことをそれぞれフィールドと呼び、横一列をレコードと呼ぶ。従って、ファイルは、フィールドの集まりでもあるし、レコードの集まりでもある。

リレーショナル・データベースでは、このファイルのことを関係＝リレーシヨン、フィールドのことを属性＝アトリビュートと呼ぶ。正確には、アトリビュートというのは、いくつかのフィールドの集まりを定義して、一つのアトリビュートと呼ぶ。

したがって、図10、(b)に示すようにアトリビュートの箱の中のいくつかのフィールドを定義して、その相互のアトリビュートのデータの関係（リレーシヨン）を知ることができる。

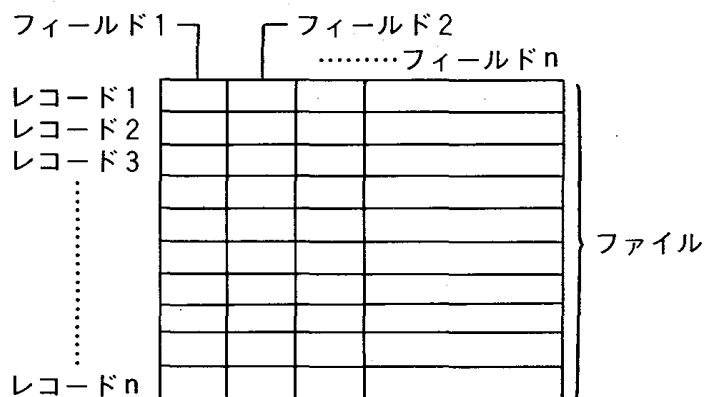
これがリレーショナル・データベースの原理であるが、このアトリビュートの関係を図11に示す3つの演算子を使って、いろいろのリレーシヨンを作表することができる。

- 1) 選択 (SELECT) という演算子を使うと、全集合のなかから、横の部分の集合を作ることができる。

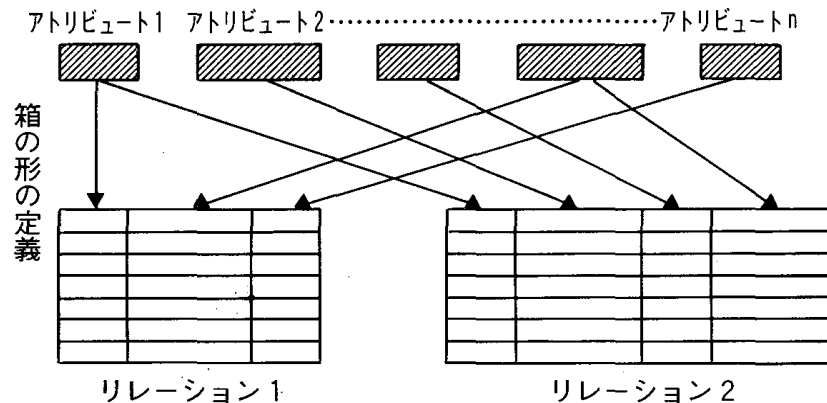
- 2) 射影 (PROJECT) を使うと、全集合のなかから、縦の部分の集合を作ることができる。
- 3) 結合 (JOIN) を使うと、別々のリレーショナル・データベースの結合ができる。

図10 ファイルとリレーシヨンの概念

(a) ファイルの概念



(b) リレーシヨンの概念



これらの演算子は、キー（顧客コードや商品コードなど）によって、選択、射影・結合がなされる。以上は、リレーショナル・データベースの原理的な仕組みで、これだけ知れば、以下の理解ができるであろう。

データベース設計によって生じる問題を突き詰めると次の3つに集約できる。⁴⁾

- 1) 応答が遅い、処理性能が低い。

- 2) アプリケーションの開発、保守の効率が悪い。
- 3) データベースの整合性が徐々に失われる。

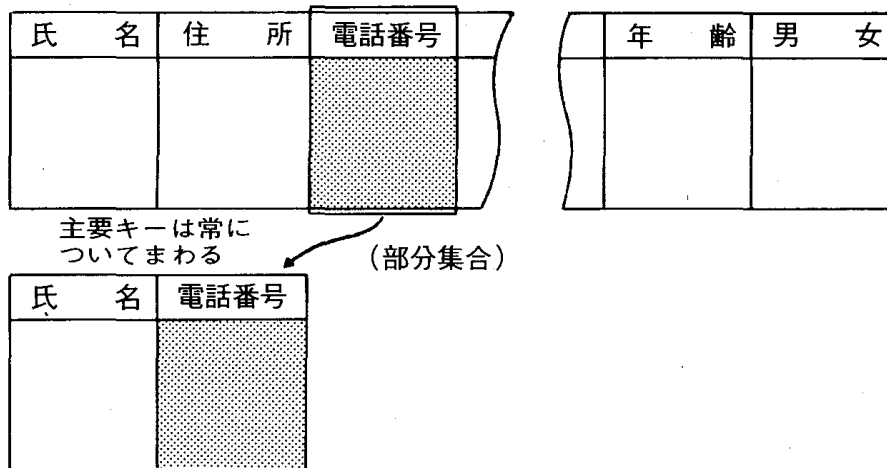
図11 リレーショナル・データベースの主な演算子

(a) 選択 (SELECT) の例



(b) 射影 (PROJECT) の例

顧客マスター・リレーション (全集合)



(c) 結合 (JOIN) の例

顧客マスター・リレーション

| 顧客No. | 氏 名 | 住 所 | 購入商品No. |
|-------|--------|--------|---------|
| 11265 | 和多田作一郎 | 東京都杉並区 | 62165 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| ⋮ | ⋮ | ⋮ | ⋮ |
| ⋮ | ⋮ | ⋮ | ⋮ |
| ⋮ | ⋮ | ⋮ | ⋮ |

商品リレーション

| 商品No. | 商品名 | 商品分類 |
|-------|-------|------|
| 62165 | 男子スーツ | 洋服 |

キー
(商品
No.)

この3つの問題は、突き詰めるとデータベースの正規化理論を無視することから生じる。米国のIBM社(当時)のE. F. Codd氏が、1970年にリレーショナル・データベースの理論的基礎となる操作体系(関係演算)を提示した。あるテーブルから所望の結果データを得るための操作として前述の図11に示す3つの演算と更新を含む5つの独立の演算を考えた。この演算を実現する言語(4GL)として「SQL」がその後作られた。データベースに記憶するデータを以下に示す正規形にすることを正規化という。これには、第一正規形から第五正規形までである。

RDBMSの理論的基盤となっているリレーショナル演算は、第一正規形でないリレーションには適用できない。現実には、どんなデータもテーブル形式に外観だけ整えれば、RDBMSで扱うことができるが、それらは厳密な意味での第一正規形ではない。SQLの規格は、テーブルが第一正規形であることを要求してはいないが、前述の3つの問題を解決するには、厳密な第一正規形が必須の条件である。

第一正規形は、次の大前提を満たしたうえで、以下の3つの条件が必要である。

大前提：各列には密接な関連がある。これは各列は意味的に結び付いている必要があるということで、実社会の関係を反映しているテーブルでなければならない、ということである。

1) 入れ子のテーブルをもたない。

すべてのテーブル要素が、行と列で特定されなければならない。一般に一つの列が複数の項目をもつ入れ子構造のテーブルは表と呼ばれるが、これは第一正規形ではない。また、複数の要素を纏めた列を含むテーブルや列の繰り返しが存在するのも第一正規形ではない。

2) 各行は重複しない値をもつ。

リレーショナル演算の基礎は集合論である(集合の元は、重複が許されない)ためにでてくることである。一つのテーブル内に同じ行はない、ということの意味する。

3) 主キーは空価をとらない。

主キーとは、各行を特定する列または列の組合せである。前の条件から第一正規形は必ず主キーが存在する。(空価とは価のない状態のことをいう)

正規化をすすめてデータベースを設計すると、アプリケーションを変更したり新たなアプリケーションを追加してもリレーショナル演算を用いたビュ一定義をすることで、既存のデータベースへの影響を最小限にすることができる。これをデータベースの独立制という。

正規化をしてないと、更新異常(更新操作によってテーブルの整合性が失われること)となり、前述の3つの問題が発生する。

正規化の理論は、性能を追及する設計者にとり強力な武器である。正規化理論とリレーショナル演算の理論は、経験と勘で設計することから救い、データベース設計を工業的な技術に引き上げることになる。このようにして、RDBMSを正しく使えば、従来の索引付きのファイルなどに比べて格段に信頼性の高いデータベースの構築ができる。このため開発者は正規化やここでは説明を省略するが、「トランザクションの概念」や、「一貫性制約、整合性制約の概念」を理解して、RDBMSの特徴を生かした設計をすることが必要である。

d. ウェアハウス・データベースの登場

経営情報システムは業務系システムと情報系システムに分けて考えるべきことを、3, (1)で述べ、図2「C/Sシステム活用領域の進化」(324頁)に示した。前述のリレーショナル・データベースは、業務系システムのOLTP(オンライン・トランザクション処理)向きのデータベースとして発展してきた。これは更新処理を容易にするためのもので、このためにデータの正規化が必要である。

これにたいして、「誰が、何時、何をして、その成果はどうなったか」という経営・管理の日々発生する生データを蓄積する「データの貯蔵庫」とでもいうべきウェアハウス・データベースが登場した。この目的は、DSS(意思決定支援システム)で、従来の定型的な集計データの検索処理ではなく、

ダイナミックに変化する単発的で、複雑な検索処理向きである。

江戸時代の商家はすべての取引を大福帳に記入していた。販売の記録簿である大福帳は、同時に経営活動のすべてを記録するものであった。したがって、大福帳を見れば、過去からの商いの動きや、取引先との付き合いがすべて分かる。このようなことを可能にするのが、ウェアハウス・データベースで、「日経ストラテジー」誌はこれに大福帳システム（大福帳型データベース）と命名している。

現在、バブル景気崩壊の嵐を受けた企業が、リストラクチャリングやリエンジニアリング（BPS）に取り組んでいるが、先進的な情報システムが経営のリストラを促進したという例はない。これまでのコンピュータシステムは「堅い情報システム」で経営活動の柔軟性を拘束したからである。

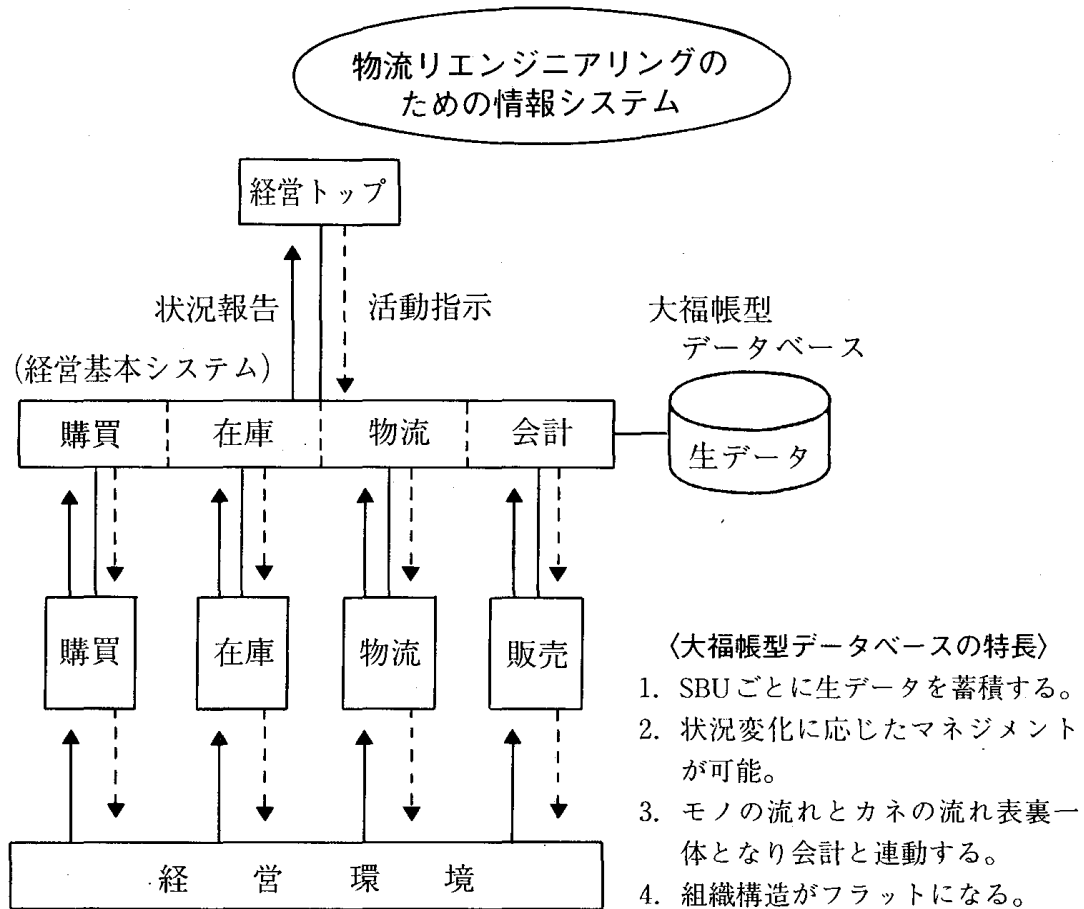
リストラ、リエンジニアリングをする場合は、情報システムの再構築から入らないと、中間管理職が減らないし、組織のフラット化はできない。

これまでのホスト中心の集中システムは、「堅い情報システム」となり経営の基本システム（モノの流れのシステム）の部門ごと（購買システムや販売システムなど）の処理で、部門データ処理を目的としていた。経営・管理のための問題の発見やその解決のための意思決定を可能にするものではなく、定型的な帳票を出力するに過ぎなかった。

C/Sシステムの普及は、分散処理とリアルタイム処理を可能にしたが、サーバーのデータベースからデータをとりだしパソコンの作表ソフト上で処理し、経営支援のための非定型的な情報の創造と蓄積を可能にし、経営・管理の問題の発見と解決のための意思決定情報を可能にした。しかし、これは限られたもので、しかも部門毎の情報処理で、かえって組織の壁が情報の管理の流れを阻害し、部門最適化を可能にしたが、全社的な最適化を可能にする情報の提供ができるものではなかった。

膨大な投資をした情報システムのこのような状況をみて、経営者の多くは怒を感じている。ここにリエンジニアリングにより組織の壁を取り除いて、ビジネス・プロセスの一貫的な流れを通して問題の発見可能な新しい情報シ

図12 リエンジニアリングのためのウェアハウス・データベースの活用



システムの構築が求められる理由がある。

図12にこの目的でウェアハウス・データベース（大福帳システム）を活用して、購買（仕入れ）から在庫・ピッキング、物流（配送）、など、ビジネス・プロセスを一貫的に流れるSBU（戦略事業単位）ごとに生データを蓄積するデータベースを示している。このシステムは、企業活動で生じたデータを「そのまま形で加工（集計）せずに蓄積」しているためにビジネス活動の実体をつかむことができる。個々のデータは、「いつ」、「誰が」、「何を」、「処理したか」まで記録するので、その活動がビジネス・プロセスを通して時間の流れで把握することができる。

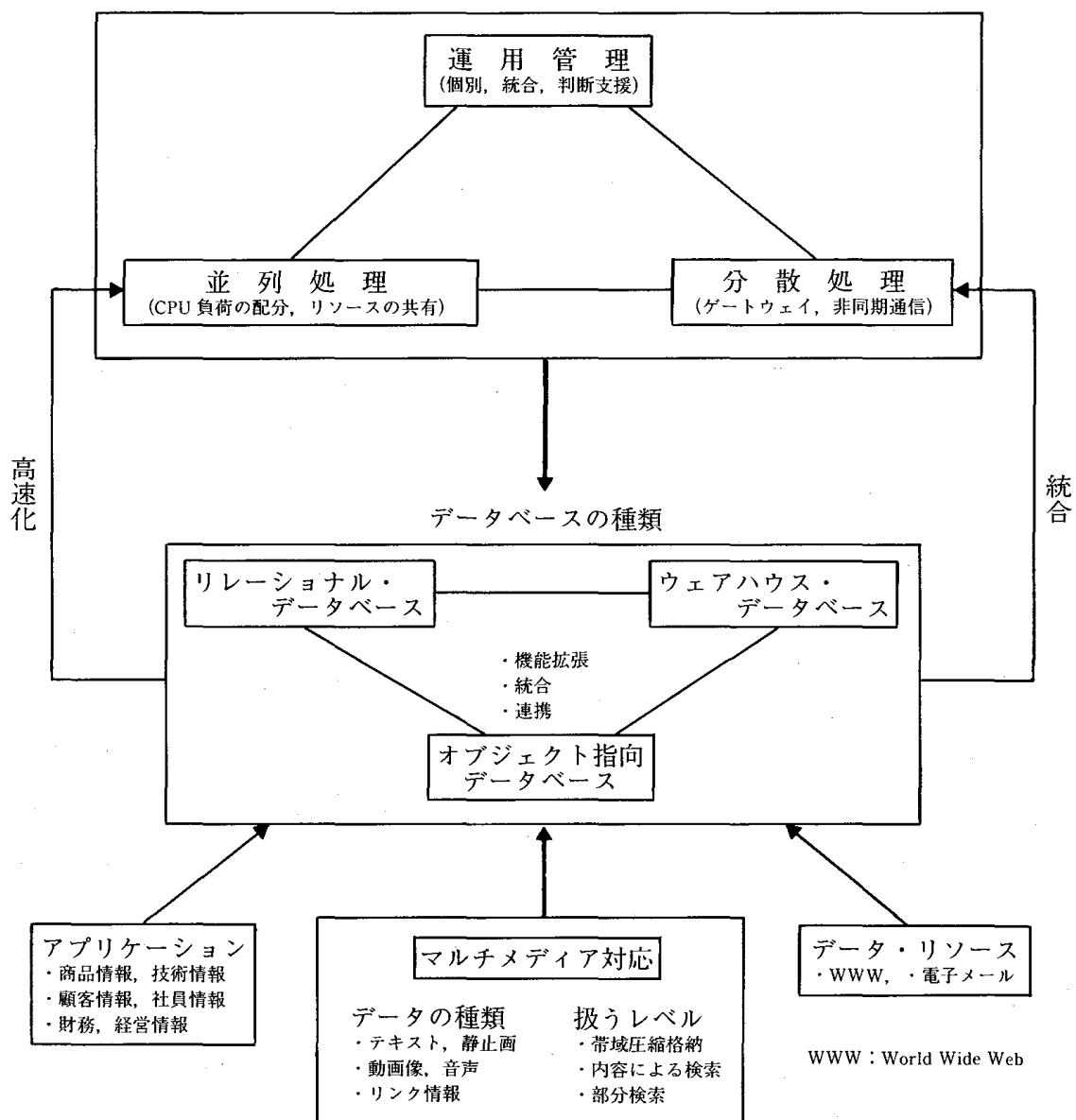
SBU単位ごとに取引データが蓄積され、発注→入荷→物流→販売という企業活動の因果関係を一体として、生データが蓄積され、それを履歴管理と

して活用できるので、そのプロセスの改善に極めて有効な情報の提供が可能となる。

e. データベース技術の進歩の方向

図13に今後の「データベース技術の進歩の方向」を示す。その概要は図示しているので説明は省略するが、データベースの種類としてリレーショナル・データベースとウェアハウス・データベースが現在、実用化されているが、今後は後述するオブジェクト指向データベースが実用化されと考えられる。

図13 データベース技術の進歩の方向



処理形態も並列処理，分散処理さらにマルチメディア対応と進展するであろう。さらに処理能力の高速化は著しく，100から1000にも及ぶCPUによる並列処理，GB（ギガバイト）単位のメインメモリーを搭載し，並列RDBMSや分散RDBMSが実用化されている。

GBクラスのメインメモリーのもたらす効果は革命的である。この例を前述のウェアハウス・データベースで説明すると，大手小売業は一日当りPOSデータの件数は数百万件以上もある。1データ当りのレコードサイズを仮に100バイトとすると総データ量は毎日数百万MBとなる。

米国最大級の小売業であるウォルマートは，数年前から秘密裏に進めてきたシステムが最近明らかになった。並列DBサーバー（NCR3600）とDB専用マシン（テラデータ）を用いて過去1年3ヵ月（65週分）のPOSの生データを蓄積している。並列DBサーバーを使うと100GB単位のDBを全件検索しても，条件によるが10秒から30秒で可能という。

一般の企業が売上POSデータを集約して蓄積しているのとは大違いで，ウォルマートは売れ筋商品，死に筋商品の発見よりも，何故売れたか，何故売れなかったかの，売れ方に重点をおいてPOSデータを活用している。これにより商品回転率を年間7回転から16回転に倍以上もよくすることができたという。

6 オブジェクト指向環境の進化

(1) 経営要件とシステム要件

生産性の高い経営情報システムを構築する場合に必要な能力は，企業の情報システムに対する仕様を明確化する能力である。経営戦略の明確化は当然経営者の役割であるが，これを経営要件の明確化ということにする。この明確化された経営要件を理解することによって，経営情報システムの要求仕様を明確化することができる。これをシステム要件の明確化ということにする。

この情報要件によってシステム要件を明確化する作業はシステム化の上流

工程である。この上流工程（システム計画や要求分析）があいまいなまま下流工程（システム設計やプログラミング）を実施すると、何回も上流工程に逆戻りして、ムダな作業が増え、コスト増、納期遅延などの問題が発生した。しかし、「4 これまでのコンピュータとパソコン LAN との構築手法の変革」で述べたように「(4) スパイラル型の三段階のシステム構築」は、データベースの設計と流通ソフトとの組合せでスパイラル（うずまき）的にシステムを構築するために、これまでの上流工程から下流工程まで一気貫通に「ウォーターホールの（滝のように）」システムを構築するのではなく、ユーザーと対話をしながらシステム化ができるのでユーザー・ニーズを満たしたシステム構築ができる。これはオブジェクト指向のアプローチである。3, (2)「Windows 環境の出現」で、「c. イベント駆動型プログラミング」を述べた。これもオブジェクト指向のプログラミングである。

「5 クライアント／サーバー・システムの構成要素の進化」で図7を示して、「クライアント／サーバー・システムの世代変化」を図示して、これからは「分散オブジェクト型 C/S システム」に移行するであろうと述べた。これからの経営情報システム構築の傾向はオブジェクト指向の流れであることを認識しなければならない。

(2) オブジェクト指向のアプローチ

オブジェクト指向の考え方とは、人間の思考方法や、現実に見ている状態そのままをコンピュータのソフトウェアに写し取ろうというアプローチである。このようにすると、完成したソフトウェアは誰でも理解できるし、別のサブシステムのソフトウェアとの整合性もとりやすくなり、メンテナンスも容易になる。

これを分かりやすく説明すると、人間は、あるモノが欲しくなると、まずそのモノを売っている商店を頭に浮かべ、そこに行って目的の商品を買うであろう。野菜を買うのであれば、スーパーの野菜売場へ、肉であれば肉売場に行く。買う場合は商品の代金をお金で払う。

この一連の行動を分析すると、肉売場で買う場合も、野菜売場で買う場合

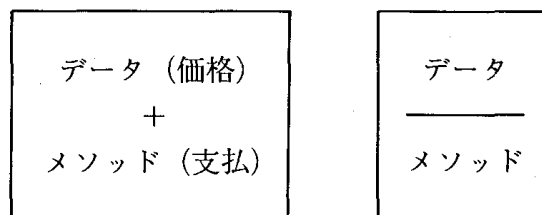
も、商品価値相当額のお金を払う、という行動は同じである。図14(a)に示すように、このデータとメソッド（方法や手続き）を一体化することを「カプセル化」といい、このカプセル化されたものを「オブジェクト」と考える。

（オブジェクトは現実の世界のモノを表すが、このようにカプセル化したものにも同じ言葉を使用する）。

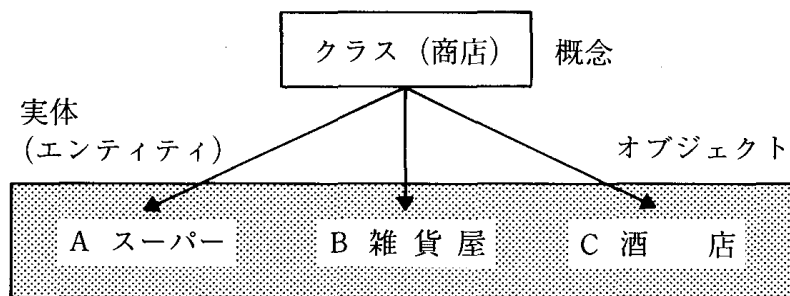
こう考えると、カプセル化されたオブジェクトは、肉を買う場合も野菜を買う場合も、その他すべての商品を買う場合に使うことができる。このオブジェクトをソフトウェア部品として使えば、そのコードと部品の再利用ができる。これまでも、ソフトウェアを部品化して再利用することが試みられたが、どんな部品があるか、その部品はどのようなものかを細心の注意を払って調べなければ使うことができなかった。

図14 オブジェクト指向の考え方

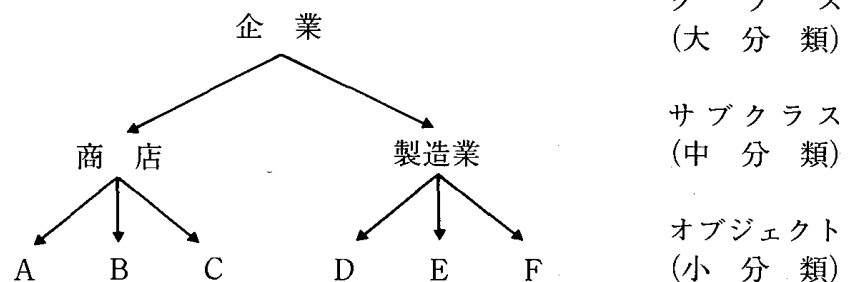
(a) オブジェクトのカプセル化（データと手続きの一体化）



(b) 概念（クラス）と実体（エンティティ）



(c) 上位概念の継承



オブジェクト指向のこの考え方には、さらに大きなメリットがある。それは図14(b)に示すように、概念と実体という考え方である。われわれは物事を見てそれを概念で覚えている。前述の買い物をする場合に商品を買う、ということを知っていて、その買う商品に最適な商店の名前を思い浮かべるのである。これは、人間は物事を固有名詞で覚えているのだが、それを概念という上位クラスで分類して覚えているということを意味する。

その分類は、図の(c)に示すように、大分類、中分類、小分類というように階層構造をしており、上位概念が継承されている。これを利用すると、大分類に相当するプログラムを作成すれば、それ以下のプログラムは改めて作らなくてもよいということを意味する。

この考え方に沿って、なるべく上位概念に相当する経営要件を明確化して、システム要件を明確化する。つまり、上流工程からこのオブジェクト指向アプローチを活用すると、システム構築が効率化する。

(3) オブジェクト指向ソフトの進化

ソフトウェア開発の生産性の向上に大きなパラダイムの変革を招来しつつある。その一つがオブジェクト指向のソフトの出現であり、もう一つはその考え方を継承した流通ソフトの組合せによるアプリケーション・ソフトの開発である。

図14に示したように、オブジェクト指向言語は、少なくとも4つの条件を備えている。(a)オブジェクトのカプセル化、(b)概念と実体：人間が考える概念と実体に基づいて個々の対象、物事を分類しクラス分けをする。その中に現実のシステム対象を入れていく、(c)上位概念の継承：大分類のプログラムを書けば、下位プログラムがその中に含まれるので、他のサブシステム（別の下位分類）のプログラムを書く必要がない。そして(d)はアクセス方式の明確化の機能で、メッセージ駆動型、あるいはイベント駆動型（イベント・ドリブン）が可能になる、ということである。

アクセス方式の明確化とは、これによってプログラムの独立性を高めるとともに、インターフェースを統一して、他のプログラムからの操作を可能に

する。これをより解かりやすく説明すると、商店では商品に値段をつけて金銭と交換するという方法が確立され、どの商店でも顧客が同じ手順で商品を買うことができる。このようなことをプログラム上で実現できるのは、メッセージ駆動（あるいはイベント駆動）でこれによりソフトの部品化を推進することができる。

オブジェクト指向言語として、最初に開発されたのは人工知能の開発に用いられた Smalltalk がある。このほかに、Prolog やリスト処理言語 Lisp にオブジェクト指向機能を付加したものが考えられた。今後よく使われるオブジェクト指向言語は、米国バークレース社の開発した C++ である。これは C 言語を拡張したもので、C 言語ユーザーに広がり、事実上のオブジェクト指向言語の標準となりつつある。

(4) オブジェクト・イネーブリング・システム

ソフトウェアの部品化と再利用のためにオブジェクト指向技術が注目され研究されてきた。Smalltalk や C++ のクラスライブラリによる部品化や再利用は、一部の成功しかおさめることができなかった。この理由は技術の修得が困難で一部の開発者しか利用できなかったことと、開発ツールが十分でなく、ソフトウェアの設計、開発、保守などにおいてオブジェクト指向のメリットを生かしきれなかったからである。しかし、本質的な理由は、再利用のメリットが特定の OS（オペレーティング・システム）や言語などの環境に限られたためである。

つまりある目的で作られたクラスライブラリーは、他のクラスライブラリーのソフト部品と組合せて使うことができないし、また他のプラットフォーム、他の OS 上では使用することができなかったからである。また、単一のアドレス空間しか動作しないので、その一部の機能を遠隔ノードで実行することはできない。したがって、これからのマルチ・プラットフォームや分散処理環境の処理体系にはこれまでのオブジェクト指向技術は力不足である。

これからのオブジェクト指向は、プログラミング環境で利用されるだけでなく、エンド・ユーザーが利用するシステム全体で操作できるものでなけれ

ばならない。このためシステム内で利用されるデータに関する統一された規格作りも必要である。これは5, (3)「(a)データ中心のシステム設計」で述べたデータ定義を厳密に行うことである。前者に対する解決法は、ソフトウェアをソース・コードレベルでの部品化ではなく、バイナリ・レベルの部品化を行うことである。これによりコンパイルなどの作業を必要としないので、利用者はプログラムだけでなくエンド・ユーザーに拡大することができる。

このようなシステムを、マイクロソフト社ではオブジェクト・イネプリング・システムと呼び、オブジェクト指向プログラミングと明確に区別している。この両方の違いを下表に示す。

オブジェクト・イネプリング・システムが完成すれば、ソフトウェア部品を組合せるだけでアプリケーション開発が可能となる。目的にあった流通ソフト部品の選択を誤らなければエンド・ユーザーそのものでもカスタム・アプリケーションの作成が可能となる。

表1 オブジェクト指向プログラミングとオブジェクト・イネプリング・システム

| | オブジェクト指向プログラミング | オブジェクト・イネプリング・システム |
|--------|-----------------|--------------------|
| 適用範囲 | プログラミング環境 | システム全体 |
| 部品化の言葉 | ソース・レベル | バイナリ・レベル(機械語) |
| 対象ユーザー | プログラマ | エンドユーザー |
| ソフトの質 | プログラマの質に依存 | プログラマの質に依存しない |

用語解説

• API

Application Programming Interface の略。OS や RDBMS などの基本ソフトウェアが、アプリケーションにサービスを提供する場合のプログラミングの規約。

• イベント駆動の開発

プログラム記述の際に、ある「イベント」(ユーザーがボタンをマウスでクリックした。あるいはデータが更新された、など)が発生したときに、実行する処理を並列的に記述していく開発手法である。これまでのプログラミングは逐次的に処理を記述するのに比べ、処理の順番をあまり気にせずプログラミングができる。対話型の GUI (グラフィカル・ユーザー・インターフェイス) アプリケーションと相性のよい開発で、ビジュアル開発ツールには常識的な開発手法である。

- ビジュアル開発ツール

極力コーディングをしないで、アプリケーションの仕様、画面、帳票のイメージなどを画面上で指定することによりアプリケーションを自動生成するツールのことである。代表的な製品は Microsoft Access, Pradox for Windows, SQL-Windows などがある。

- クラス・ライブラリー

オブジェクトの持つデータや処理（振る舞い）を規定するテンプレートの集まりのことで、Smalltalk, C++などのオブジェクト指向言語で必須の概念がクラスである。クラスと「継承」の機能をうまく使うとアプリケーションの開発効率を飛躍的に向上できる。最近では Power-Builder, SQL Windows, Oracle Power Objects などのビジュアル開発ツールにクラス概念を取り入れている。

7 オープンシステム構築に向けての標準化技術の進歩

(1) ソフトの部品化と再利用のための技術開発

「Windows95」には「OLE2.1」と「プラグ・アンド・プレイ」(PnP と略称)と「プロパティ」という機能が導入されている。Windows3.1では「OLE1.0」であったが、2.1ではこれらの機能が盛り込まれて、ソフト部品化と再利用の機能がより強化されている。

OLE とはマイクロソフト社が提唱するアプリケーション間の連携仕様 (Object Linking and Embedding) で、“オーレ”と発音する。前バージョンの OLE1.0 では例えば表計算ソフトで作成したグラフをワープロ文章の中に張り込む、といった複合ドキュメントを作成するための色彩が強かった。これに対して OLE2.0 ではOLE カスタム・コントロール (OCX と略称) や OLE オートメーションといった概念を取り入れて、より複雑なアプリケーション間の連携に道を開いた。マイクロソフト社の Access2.0 だけでなく、Pradox 5.0J for Windows などサード・パーティからも OLE2.0 対応の製品が日本でも出荷され始めている。これにより流通ソフトの部品化再利用のための技術が本格普及しようとしている。

「プラグ・アンド・プレイ」(PnP) とは、パソコンのハード・ソフトの部品を、家電製品のようにプラグを差し込んだらそのまま使える、という意味

で、米国のソフト企業がよく使う用語である。例えば、マイクロソフト社がインテル社と共同で提唱した「Plug and Play 仕様」は、LAN 接続ボードや CD-ROM などの各種デバイス設定を自動化するための仕様である。

「プロパティ」とは、Windows95 のあらゆるオブジェクトに関する情報を見たり、さまざまな設定を行うためのものである。デスクトップ上のオブジェクトの操作が、右ボタン操作とプロパティですべて行えるようになっている。例えば、ハード・ディスクのプロパティを見ると、ディスクの空き容量を見たり、ディスクの最適化などのツールの実行、ネットワークの共有設定が行える。

これからのパソコン活用は、エンド・ユーザー・コンピュータに向けて、流通ソフト部品を組合せて、アプリケーションまで自由に作成できる方向を指向している。

このため OLE とオートメーションという概念について理解しなければならない。このために、3, 「(2) Windows 環境の出現」を再読して、Visual Basic や VBX を理解して以下に進めたい。Visual Basic のカスタム・コントロール (VBX) などのいわゆる「コンポーネント・ソフトウェア」が登場している。このコンポーネント・ソフトウェアは、従来のオブジェクト指向のソフトウェア部品よりも、機能レベルが高く使いやすい部品と位置づけられている。

特に、Visual Basic には、サードベンダーから各種の VBX が提供されており、米国では新しいソフトの流通市場を確立した。これによりソフト部品の再利用の道を開いた。データベース・アクセス、通信機能、電子メール、FAX 通信、画像処理、3次元グラフィックス、表計算などさまざまなものがある。このようなコンポーネント・ソフトウェアの登場の意味することは、Visual Basic のような非オブジェクト指向言語でもソフトウェアの部品化が可能であることを示した。

しかし欠点は、VBX は、Visual Basic と Windows という OS の環境でしか動作しないためマルチプラットフォームや分散環境での適応はできないこと

である。

(2) ソフトウェア開発を変革する OLE

マルチプラットフォームや分散処理環境でソフト部品を再利用するためには、前項「6 オブジェクト指向環境の進化」の(4)で述べた、「オブジェクト・イネーブリング・システム」を実現して、バイナリー・レベルでソフトの部品化を実現することである。このソフトの部品化を実現する規格として、現在最も有力視されているものは、マイクロソフト社の OLE である。それはこれまで全世界で 6 千万本以上の出荷実績を誇る Windows の標準規格であり、さらにこれから普及する Windows95 にも採用されているからである。さらにワープロ・ソフト、表計算ソフト、グラフィック・ソフトなど多数の OLE 対応アプリケーションが既に実用化されているからである。これにより、今後、OLE がソフトウェア部品化規格のデファクト・スタンダード化する可能性が非常に高いからである。

OLE は、マイクロソフト社にとって、ソフトウェアの部品化という概念を提供するだけでなく、Windows の基礎技術と位置づけている。OLE は Windows 環境以外にも広がっている。米国で出荷した「Microsoft Office」は、マッキントッシュ版を出荷し、日本でも Excel のマッキントッシュ版が出荷されている。さらに UNIX 上にもこれらを移植するとマイクロソフト社は声明している。これらのアプリケーションは、すべて OLE2.0 に対応していて、将来、OLE は OS の壁を破りオープン・システム化を主導することが期待されている。

米国で販売した「Microsoft Office for Windows95」(Office95 と略称)は従来の Office に入っていた Word, Excel, Power Point, Access に、スケジュール管理ソフトの Schedule+を加えて一つのパッケージにしたものである。これには、Windows に標準搭載された OLE2.0 を駆使している。これには「Office バインダー」という機能がある。これは複数のドキュメントをバインダーという一つのファイルに纏めて管理するものである。例えば、一つのバインダーに Excel で作成した文書を入れておくことができる。バインダー

にのれるドキュメントは OLE のオブジェクトで、OLE2.0 に対応しておれば他社のアプリケーションのデータも入れることができる。

このように OLE はアプリケーション間の連携を可能にする。さらに Office95 には、Windows が標準搭載した、通信・ネットワーク機能を利用することができる。その一つは、パソコン通信サービスの「The Microsoft Network (MSN)」との連携ができる。Office ショート・カットバーには MSN というボタンがあり、そこに常時利用するフォーラムなどを登録することができる。そのボタンをクリックするだけで、MSN にアクセスしてフォーラムに入ることができる。インターネットへのアクセスも Word のアンドオン・ソフトとして「インターネット・アシスト」が用意されている。これを使うと、Word の中から直接インターネットの WWW サーバーにアクセスできて、その情報を Word の中に取り込むことができる。

OLE は 1 台のパソコン内の幾つかのアプリケーション間の連携を可能にするばかりでなく、分散ネットワーク上のアプリケーションとの連携も可能となる。

(3) OLE 対応のソフトウェア部品化環境

OLE が実現するソフトウェア部品化環境は、表 2 に示すように、大きく分けて「コンテナ」と「サーバー」に分かれる。コンテナは、ソフト・コンポーネント間の連携をするための複合ドキュメントの台紙に相当する。サーバーはソフト・コンポーネントに相当する。これらのソフト部品間の通信を

表 2 OLE を構成するソフト部品の構成

| コンテナ(例: Visual Basic3.0等) | | サ ー バ ー | |
|----------------------------------|---------------------|---|--------------------------------------|
| サーバーを貼り付ける部品 (複合ドキュメントの台紙に相当) | | オブジェクト化されたデータを作成する部品 (複合ドキュメントのコンポーネントに相当) | |
| スマート・コンテナ | ダム・コンテナ | OCX | OLE サーバー |
| アプリケーション 開発に利用する | 複合ドキュメント 開発に利用する | 主に GUI 部品 (In-process Server) | OLE 対応アプリケーション (例: Excel, Word 等) |

実現し、OLE にプログラマビリティを追加するためにマイクロソフト社が用意した規格が「オートメーション」である。オートメーションとはコンテナとサーバー間の通信を実現する。

オブジェクトを貼り付けるコンテナは、さらに「スマート・コンテナ」と「ダム・コンテナ」に分かれる。「ダム・コンテナ」は、オブジェクトを貼り付けるというコンテナとしての基本機能だけを備えたものである。これにより、エンド・ユーザーは、一つのアプリケーションでデータを共有する複合ドキュメントを作成することができる。

「スマート・コンテナ」は、この基本機能に加えてアプリケーション開発に必要な機能を備えている。業務システムなどのアプリケーション開発にはスマート・コンテナを選択する。

サーバーも表に示すように2種類に分かれる。「OLE サーバー」というOLE 対応の表計算ソフトやワープロ・ソフトなどのアプリケーションそのものである。もう一つは、ボタンやスクロール・バーなどのGUI 部品を提供する「OCX」である。

OLE サーバーも OCX も、コンテナに貼り込む場合は、いずれもドラッグ・アンド・ドロップ・カット、アンド・ペーストでメニューの「オブジェクト貼り込み」操作でコンテナに貼り込むことができる。

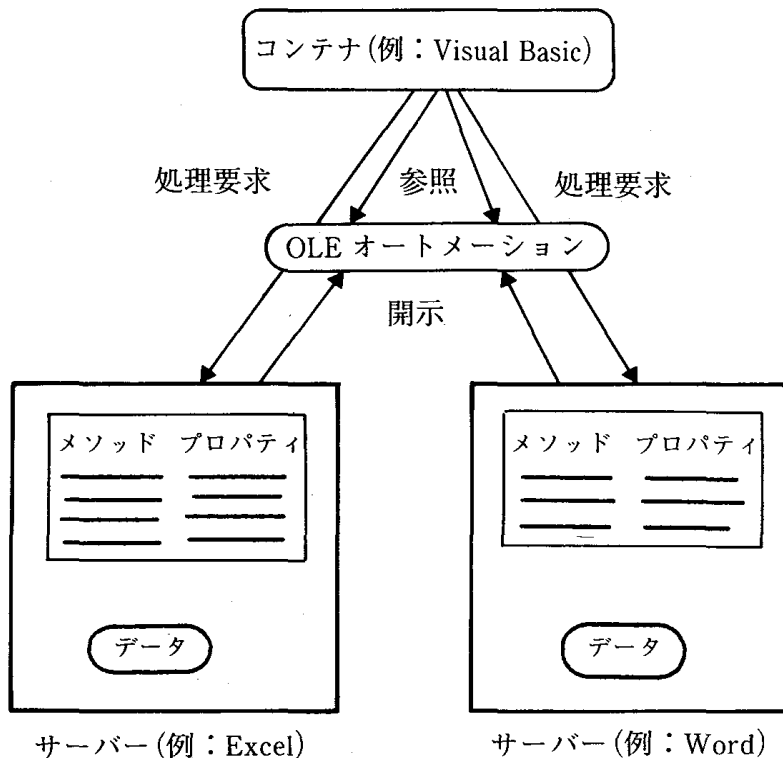
両者の大きく異なる点は、OCX はコンテナと同じプロセスで動作する「In Process Server」であり、OLE サーバーは呼びだされる度に別プロセスをメモリー上に展開する点である。したがって、OCX は常にアクティブに瞬時に動作する。

(4) OLE オートメーションの動作

図15に「OLE オートメーション」の仕組みを示す。スマート・コンテナは、オートメーション対応、つまりオートメーション・コマンドのインターブリタであるコントローラを備えている。外部からのオートメーション操作を受け付けて、自分の持つサーバーに対して、外部からのオートメーション・コマンドを受けわたす機能がある。ソフト部品の選択は、オートメーション対

応が最低限必要である。理由はサーバー間の処理を自動的に実行するためである。さらに、コンテナに関しては、OCX に対応したスマート・コンテナを選択する。複雑で大規模なアプリケーションを開発するにはスマート・コンテナの機能が不可欠だからである。

図15 OLE オートメーションの仕組み



図示するように、サーバーはプロパティとメソッドをオートメーションに対して開示する。コンテナは、それらを参照してサーバーに対して処理要求をする。「部品ソフトに対して何をさせるか」を記述するのがオートメーションのプログラミング・スタイルである。これによってプログラム全体を Visual Basic で記述する場合に比較して著しく単純でシンプルで、早くシステム構築ができる。

(5) OCX と VBX の違い

3 節, (2), e. で「カスタム・コントロール VBX」について述べた。ここで OCX の違いについて述べる。OLE とオートメーションを組合せて、ア

アプリケーション開発ができることが理解されたであろうが、OLE サーバーだけでは洗練されたアプリケーションを開発することはできない。それは使いやすい GUI が構築できないこと、これを「In Process Server」で実行できないからである。同じプロセスで順次実行できる GUI として OCX (OLE カスタム・コントロールの略) が登場したのである。

同様のソフト部品として VBX (Visual Basic カスタム・コントロール) と呼ぶ製品群が登場して、ソフト部品化への道を開いたことは前述した。

VBX と OCX は共にオブジェクト・イネーブル・システムの製品で、バイナリ・レベルのソフトの部品化で汎用性があるが、両者には大きな違いがある。VBX はオートメーション対応でないということと、それを貼り込める相手は基本的に Visual Basic もしくは Visual C++ で記述されたアプリケーションに限定されることである。また16ビットの環境の製品であり、Windows NT や Windows95 向けのアプリケーション開発用の部品としては能力不足である。

つまり、OCX は VBX の欠点を解消したソフト部品でありサーバーの備える汎用性とプログラマビリティを付け加えたもので、これからのソフト開発の主流になるものと期待されている。しかし、OCX を利用するには、OCX 対応のコンテナが必要であるが、現在、英語版の「Microsoft Access Ver2.0」がある。MFC (Microsoft Foundation Class) が提供されているが、OCX を開発するためのライブラリーは含まれていない。OLE 開発環境として現在大きく欠落している部分があるが、急速に整備されてくるものと考えられる。

備考：本稿においては OLE を述べたが、マッキントッシュにも OpenDoc という、ソフト部品を組み合わせるオブジェクトを構築するコンポーネントウェア技術がある。OLE も OpenDoc も分散オブジェクト管理の標準仕様である CORBA (共通オブジェクト・リクエスト・ブローカー・アーキテクチャ) に準拠して作られている。CORBA はネットワークで接続された複数の異なるハードウェア、OS 上のアプリケーションが連携できるようになる標準仕様である。

8 通信技術の発展と今後の展開

(1) N-ISDN から B-ISDN へ

これからの情報システムはコンピュータと通信システムを一体的に考えなければならない。1998年4月の電機通信事業法の改正による全面的な通信自由化時代の到来が始まって、1998年にはNTTによるISDNの実用サービスが始まった。これは図16に示すように現在別々の張られている通信網が、ISDNによって総合化され、これが21世紀に向けての世界の政治・社会・経済活動のインフラストラクチャ（基盤設備）になるからである。現在、先進諸国はもちろん、アフリカの山奥まで、世界中の通信網がISDNに張り替えられることが決まっているからである。

このように通信システムの今後の展開は比較的容易に推定することができる。それはITU（国際通信連盟）の下部機構であるCCITT（Consultation Communication International Telegraph and Telephone: 国際電信電話諮問委員会）で決定されているからである。

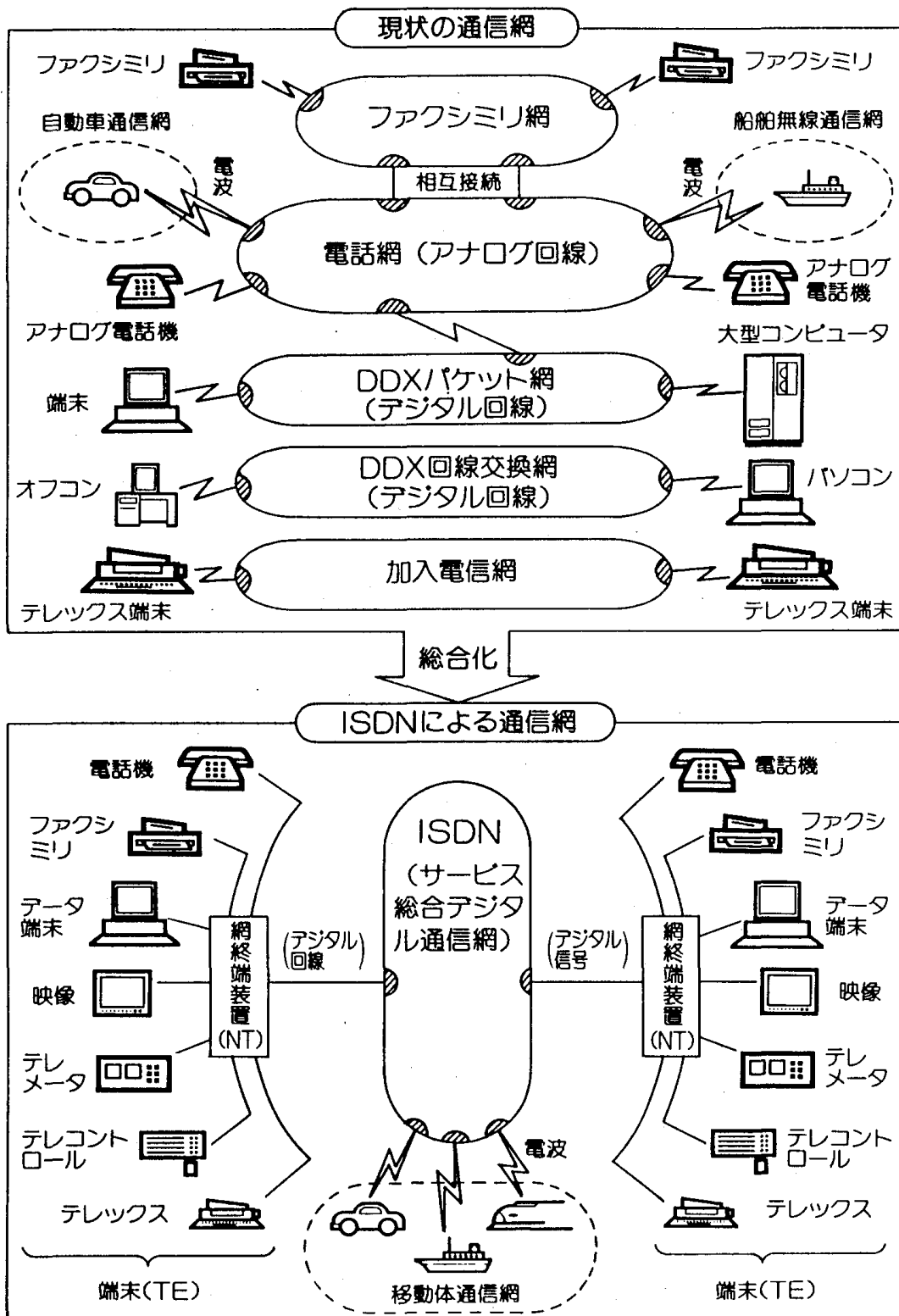
現在実用化されている狭帯域のISDNでは正確にはN-ISDNと呼ぶべきものである。（現在のISDNは64kb/sから1.5Mb/s間での伝送帯域）近年の情報処理技術の進展、各種のOA端末、特に音声やデータだけでなく、静止画、動画などのさまざまなメディアの通信が要望されるというマルチメディア時代を迎えて、1.5あるいは2 Mb/sを越える宏帯域のISDN，B-ISDNが求められるようになってきた。

(2) B-ISDN を理解する SDH 環境

N-ISDNでは64kb/sは音声信号をデジタル化して伝送する場合の世界共通の伝送速度で、これを2本あるいは幾つかを束ねて伝送している。

SDHは同期デジタル・ハイアラキー（Synchronous Digital Hierarchy）の略で、64kb/sの回線を段階的に束ねて伝送路を作るための通信系列のデジタル・データの同期多重方式の国際標準のことである。この方式が日米とヨーロッパで違っていたのを図17に示すように、1988年に155.52Mb/sを基

図16 ISDN による通信サービスの総合化



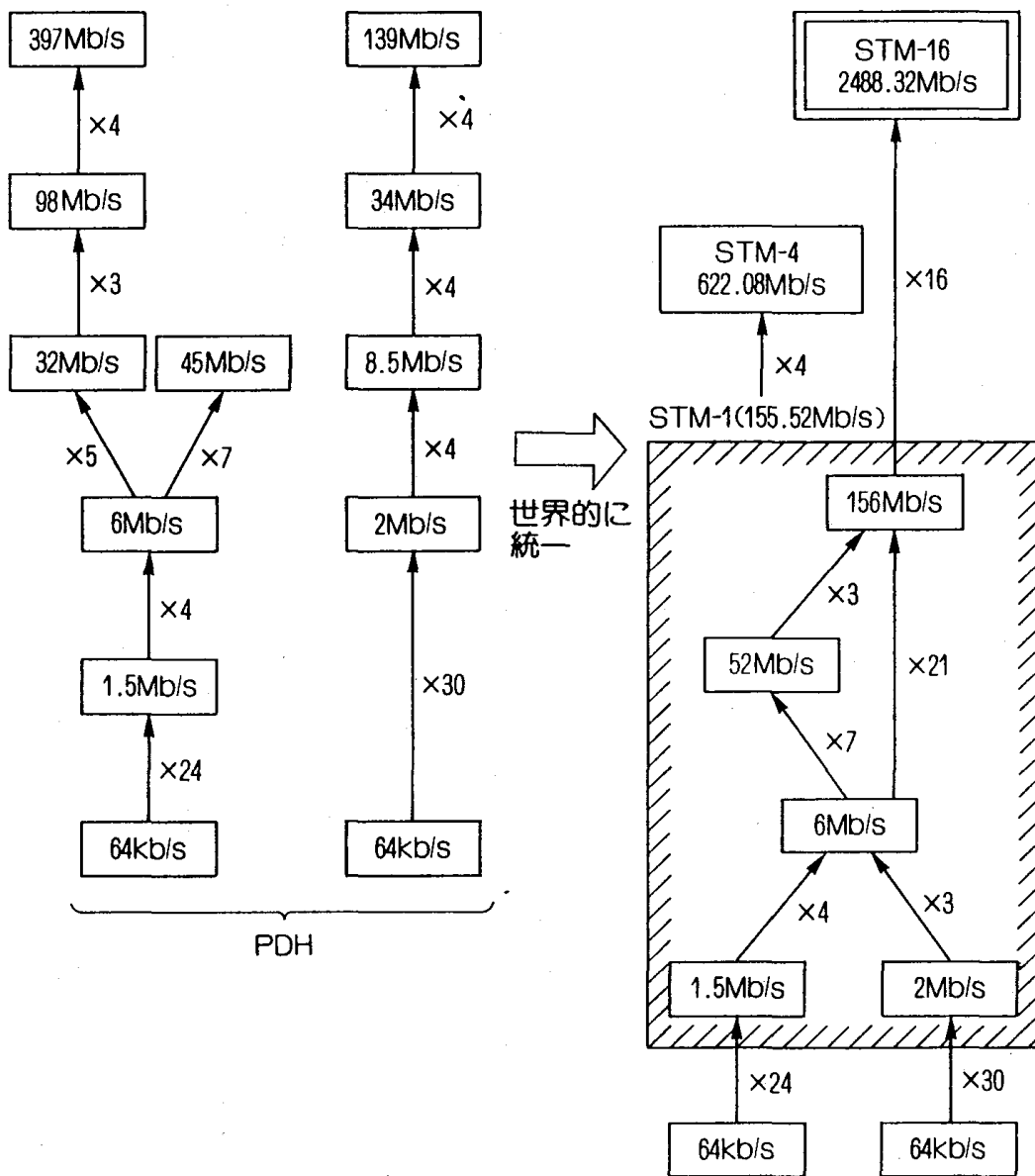
(和多田作一郎著,「新訂・ISDN の基礎を知る事典」実務教育出版刊, 17頁より転載)

図17 これまでの非同期ハイアラキーを世界的に SDH に統一

(a) これまでの日米の
ハイアラキー

(b) これまでのヨーロッ
パのハイアラキー

(c) SDHの多重化方式



(和多田作一郎著,「新訂・ISDNの基礎を知る事典」実務教育出版刊, 227頁より転載)

本とするハイアラキーに統一することが決められた。これが基本単位（レベル1）でSTM-1（155.52Mb/s）である。さらにそれ以上の高速通信も世界的に共通の尺度で実現できるように決められている。

(3) マルチメディア時代の情報ハイウェイ：宏帯域 B-ISDN

最近、HDTV (High Definition TV: 高解像度テレビ) などのより高品質な動画像の伝送や、CAD/CAM 向けデータの伝送、あるいは100Mb/s を超える高速な LAN 間の通信が求められるようになってきた。

このため、これまでの N-ISDN で提供可能な1.5Mb/s あるいは2 Mb/s を超える高速データ伝送や高品質の映像サービスが提供できるような宏帯域の B-ISDN が求められるようになってきた。

これにより、一本の回線を多重に使うマルチメディアの伝送を可能にしたうえで、これまでよりもさらに高速な信号の伝達ができるネットワークの出現が求められることは容易に想像できる。

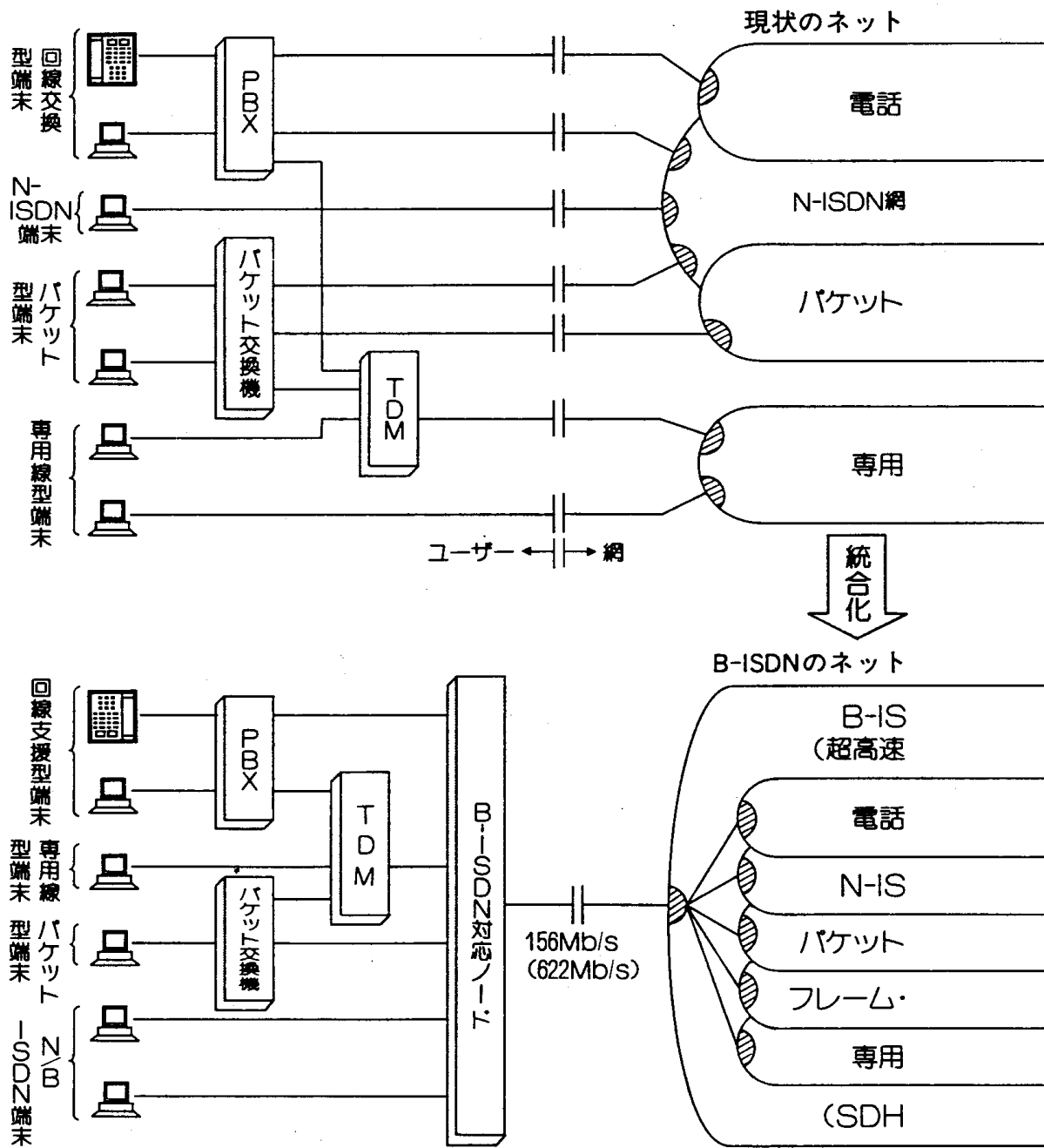
これからの企業内の情報通信システムの主役となる LAN は、信号伝送速度は高速であるが、実際に送られる情報量の通信帯域に対する密度はせいぜい数%である。このことから、LAN 間接続などのコンピュータ間通信にはこれまでの回線交換方式はふさわしくないことがわかる。B-ISDN では、このような LAN 間通信に相応しい方式が求められる。

映像信号は動きの激しい画面ほど情報量が多く、逆に静止画像の場合は情報量が少なくなる。高品質の TV 放送や HDTV では少なくとも数 Mb/s 以上の信号速度が必要である。

これに比べて、LAN 間接続の場合は、瞬間的には、1 Gb/s (ギガ b/s) もの情報を送る必要がある高速 LAN も出現しつつある。これらを両立させて、回線の通信効率を上げる通信システムとして、B-ISDN が登場したのである。

前述のことを可能にするには、通信目的に応じて低速から高速までの可変速度のネットワークを構築することが求められる。「伝送信号速度の柔軟性」と、コネクション型 (送信局と受信局間に通信路を設定してから通信をする) の通信と、コネクションレス型 (LAN のように回線に接続している局に一斉に通信をする) 方式が採用できるように、「多彩な接続機能」を持たせるようになっている。これによって、一対一通信 (相互通信型サービス)、一

図18 B-ISDN 時代におけるネットワーク構成とユーザー設備のイメージ

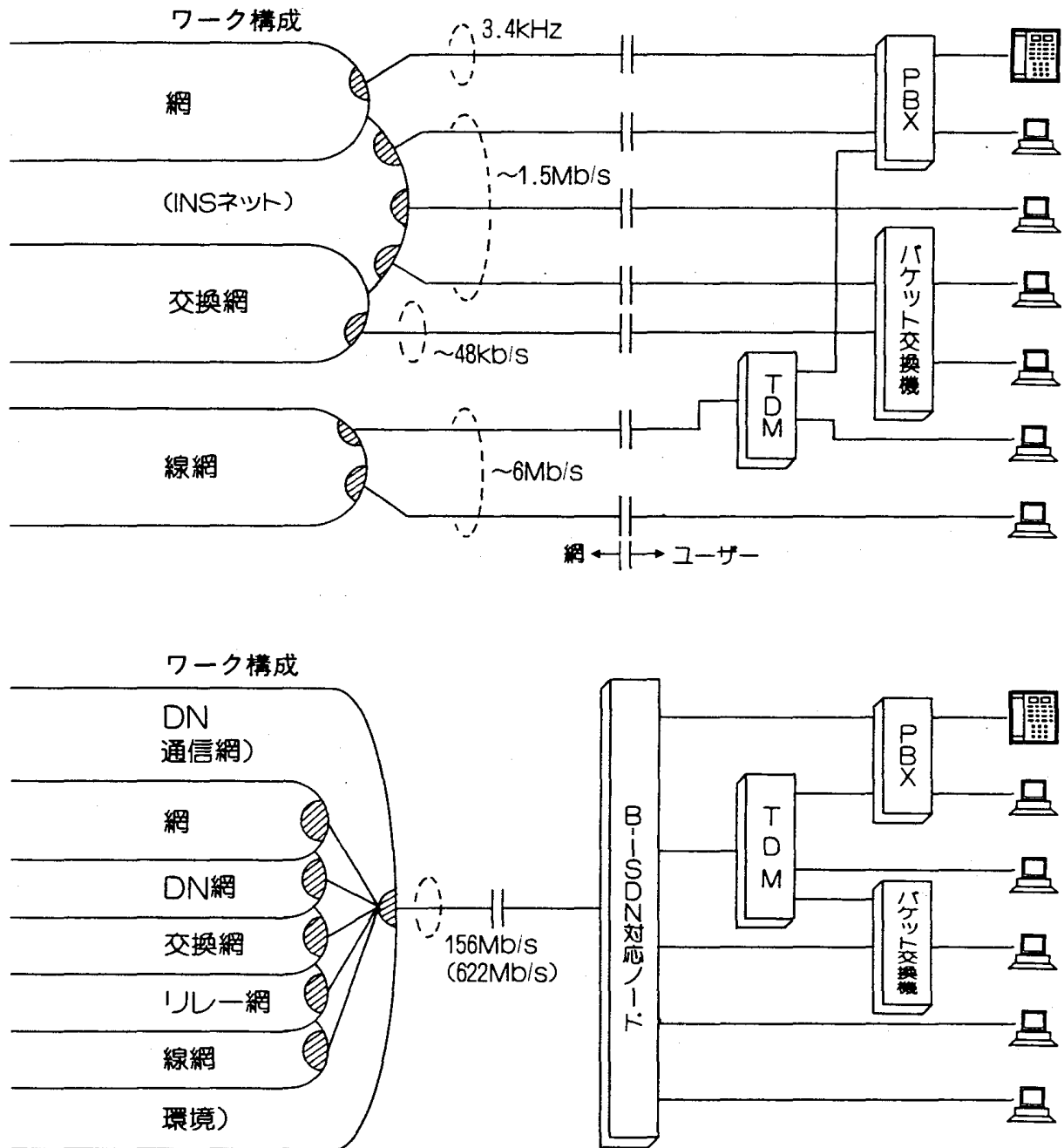


〔備考〕

—|— : ユーザー・ネットインタフェース

PBX : 構内交換機

TDM : 時分割多重化装置



(和多田作一郎著,「新訂・ISDNの基礎を知る事典」実務教育出版刊, 200頁より転載)

対 n 通信 (分配型通信サービス), n 対 n の通信 (会議型通信サービス) が自由に選択できる。

B-ISDN は公衆通信回線であるが, ユーザーによってネットワークを自社だけで占有して使用したいという要望が強い場合がある。あるいは国内だけでなく, 外国のように離れた事業所間を内線番号 (3桁程度) で通信したいという要望もある。これを実現するために, 仮想専用線網 (VPN: Virtual

Private Network) というサービスも実現する。

さらに時間によって、伝送する情報量が異なる場合に、契約回線容量や接続先をユーザー自身で変更したい場合もある。このようなサービスをエンド・カスタマ・コントロール (End Customer Control) と呼んでいる。

以上のような要望を満たす次世代のネットワークが B-ISDN である。

以上を要約すると、B-ISDN には、N-ISDN で実現できなかった次のような特徴がある。

- 1) 低速から高速までの総合的な通信サービスで、155.52Mb/s または 622.08Mb/s 速度などのインターフェイスを提供する。
- 2) 通信目的に応じて、固定速度でデータを伝送したり、可変速度で送るなどの「伝送信号速度の柔軟性」のある使い方ができる。
- 3) コネクション型の通信とコネクションレス型の LAN のような「多彩な接続機能」が可能で一对一、一对 n 、 n 対 n の通信が自由に選択できる。
- 4) 公衆通信回線であるが、必要に応じて専用回線のような使い方（これを仮想専用回線という）ができる。また、時間によって伝送する情報量が異なる場合でも、契約回線容量や接続先を変更できる、という「ネットワークの運営管理に柔軟性」がある。

図16に「ISDN による通信サービスの総合化」を示したが、これを含めた「現状ネットワークの構成のイメージ」を図18の上を示す。

本格的な高速マルチメディアへの対応は、現在の N-ISDN で提供している INS ネット64 (64kb/s) や INS ネット1500 (1500kb/s) ではスループット不足で対応できない。さらに FDDI (Fiber Distributed Data Interface) LAN を相互に接続したり、動画像 Windows を自由に利用したいという場合に伝送容量が足りない、これに対して、B-ISDN は、現在の N-ISDN の64kb/s を一気に156Mb/s、さらに次の段階では622Mb/s にまで拡張しようという構想である。

図18の下に示すように、B-ISDN は、現在の高速デジタル伝送サービス

(スーパーデジタル) を吸収するばかりでなくすべてのサービスを一回線で提供するように統合化される。

B-ISDN は高速広帯域だけのサービス専用ではない。したがって、N-ISDN は低速で、B-ISDN は高速であるというふうに区別して考えるのはその本質を理解していない。図示するように一本のアクセス回線で、これらすべてを統合化したものと考えべきである。

(4) 発展するネットワーク技術

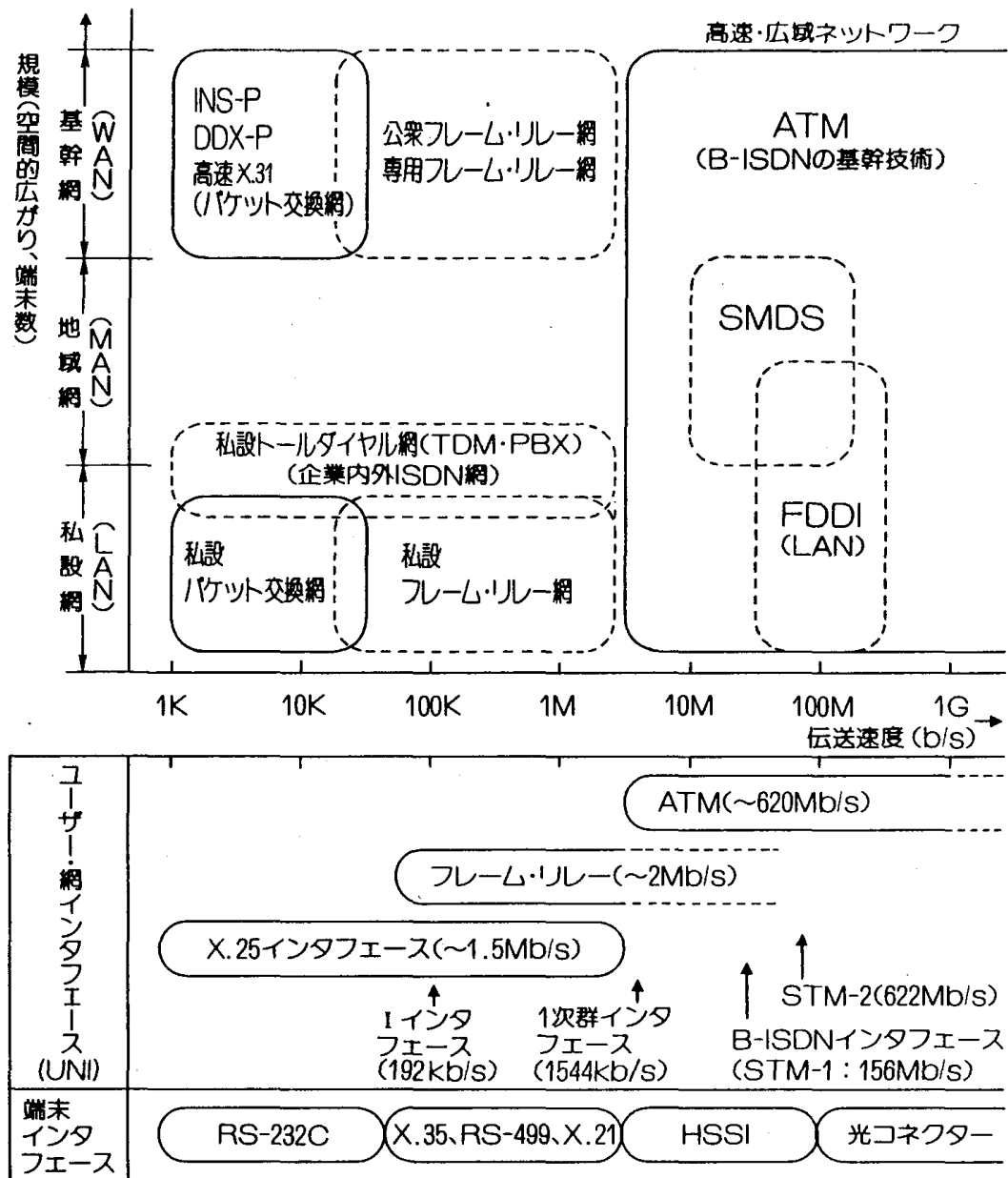
増大する通信量に対処するために、通信回線の開拓に血みどろの競争が繰り広げられた時代は光ファイバー伝送技術の開拓によって終わりをつけた。光ファイバーの登場によって、髪の毛ほどの細さ（直径0.1ミリメートル程度）のもので数十Gb/s（1秒間に数十×10⁹ビット）の情報が送れるようになったからである。これはたとえて言うなら一度に十数万人の電話で同時に通話できるほどの大容量の伝送が可能となったことを意味する。しかも、銅の電話線に比べて数倍程度のコスト差までになってきた。

光ファイバー・ケーブルにより高速データを大量に流して、回線の利用率を上げる方向に向けられていくだろう。図19は「高速、広域に向けて発展するネットワーク技術」を概念的に示した。

フレーム・リレーは、FMBS (Frame Mode Beara Service: フレーム・モード・ベアラサービス) と呼ばれていて、狭帯域サービスと位置づけられている。図は UNI とその端末インターフェースと LAN, MAN (Metropolitan Area Netwrk: 地域網), 私設網 (LAN) に区別して、より高速に発展する様子を示している。

これからの次世代の基幹網は、B-ISDN でありその基幹技術は SDH 環境の理解と、ATM (Asynchronous Transfer Mode: 非同期運送モード) であることを図示している。

図19 高速・広域に向けて発展するネットワーク技術



〔備考〕

HSSI : High Speed Serial Interface

FDDI : Fiber Distributed Data Interface (LAN)

SMDS : Switched Multimediabil Data Services

ATM : Asynchronous Transfer Mode

〔企業内外ISDN網(私設トールダイヤル網)は81~83節を参照〕

(和多田作一郎著, 「新訂・ISDNの基礎を知る事典」実務教育出版刊, 223頁より転載)

9 これからの情報化教育は？

(1) ソフトウェア産業の将来

オープンシステム化の進展と OLE（オーレ）により，ソフトウェア開発においてこれまで長い間課題であった汎用ソフト部品の流通を促進することは間違いなからう。OLE の中核技術は，ソフト部品として提供される OCX と，ソフト部品間の連携と，アプリケーション間の連携を可能にするオートメーションである。これとアプリケーション間連携のための多彩なインターフェースの標準化（例：ODBC，VIM，MAPI 等）の出現は，大規模分散処理環境におけるエンド・ユーザー・コンピューティングを可能にする。

これによって，ソフトウェア産業には2つの階層が出現するであろう。C，あるいはC++言語を活用して，流通部品ソフトであるコンポーネントを作成するコンポーネント開発者とその部品を活用して，システムを構築するシステム構築者（現在のシステム・インテグレーター）に分かれるであろう。そして，究極は，エンドユーザー自身でシステムを構築できるところまで流通部品ソフトの汎用性が広がることであろう。

(2) 企業の求めるシステム・アドミニスト・レーターの変革

以上述べたことから，これまでの通産省の設定した情報化の国家試験は大きな変革を受けることが理解されるであろう。そのなかに，システム・アドミニスト・レータ（SAD と略称）という，企業の利用部署に所属しエンド・ユーザーの立場から情報化を支援する情報リーダー育成の国家試験がある。これも大きな変革が求められる。

これからの企業が求める SAD としての能力は，次のようなものである。

- 1) 情報の共有のためのデータベース構築の知識が必要である。(5, (3)「データベース技術の発展」参照)
- 2) 流通ソフト部品の機能とその活用に関する幅広い知識が必要。
- 3) これからの企業情報システムの中核になる LAN と企業間を接続する WAN に関する幅広い技術が必要。これまでの SE（システム・エンジ

ニア) はコボル等のプログラム言語が必要であったが、これからはハード、ソフトの幅広い知識が必要となり、特にこれまで必要でなかったミドルウェアの知識が必要。

4) 特にネットワークにおけるボトルネックの回避が、システム構築における最大の注目点であり、LAN, WAN に流れるトラフィックの推定とその平準化のためにハードの知識が必要不可欠となる。

5) 経営組織と情報の関係、特に情報活用による経営の生産性の向上に関して理解して、必要な情報が必要な箇所で創造・蓄積活用することの指導が必要である。

(3) これからの情報化教育の在り方

以上述べたように、ソフトウェア産業の変革と企業の求める情報化指導のキー・パーソン (SAD) の変革は、大学における情報化教育の大変革を招く。これまでよりもより幅の広い教育が必要ということである。特にこれから高等学校における情報化教育が進展するであろうから、これとの連携をどうするか、ということも主要注目点である。EXCEL 等の作表ソフトはある程度高等学校で学んでくるであろう。その基礎の上に連携して、より高度な流通ソフト部品の組み合わせにより OLE オートメーションを活用して、自由にアプリケーションの構築できる教育が必要である。

さらに、「情報の共有」と「情報活用と蓄積」のためのデータベース理論は経営情報システムの中核になるので、その種類とデータ更新の仕組み、トランザクション・エラーの対策等までも含めた知識が必要である。

特に LAN, WAN 等のネットワークの知識はハード、ソフトを含めてこれからの経営情報システムを中心技術として必要となる。

現在、通信技術者が非常に不足している。そしてこれまで教育を受けたコボル文化時代の SE が、通信技術の不足のために役に立たない状況にあることも考え、通信システムを中心にした幅広い情報通信システムの知識を教育する必要性を痛感する。

10 あとがき

富士山の壮麗さは、裾野がバランスよく広がり日本一高いからである。10年後の役立つ企業人にする情報化教育は、富士山のごとく基礎教育を重視して幅広い知識を持ち、しかも経営学に専門化した日本一優秀な人材の養成になるのではなかろうか。上記を叩き台にしてご論議をいただき、本校が他校と差別化して発展するための情報化教育のカリキュラム編成に役立てば筆者として望外の喜びである。

注

- 1) 竹内郁雄稿「あたためも八掛」『情報処理』, (社)情報処理学会刊, 1991年1月号, 11頁。
- 2) 「ビジュアル開発」, 『日経オープンシステム』, 1993年7月号, 図2を参考。
- 3) 「一から学ぶC/Sのソフトウェア構成術・ミドルウェアを制すればC/S構築は百戦危うからず」, 『ネットワーク・コンピューティング』, 1995年, 6月号, 33頁。
- 4) 「AccessとVBユーザーのためのDB入門(1)」, 『日経オープンシステム』, 1994年, 10月号, 286頁。
- 5) 「基幹系へ適用領域を広げる」『日経オープンシステム』, (日経BP社刊) 1994年4月号, 112-127頁を参考に作成。

参考文献

- (1) 小田島孝久稿「ソフトウェア開発を変革するOLE」, 『日経オープンシステム』(日経BP社刊) 1995年5月号, 37-316頁。
- (2) 「大規模C/Sビジュアル開発」, 『日経オープンシステム』(日経BP社刊), 1995年3月号, 232-257頁。
- (3) 和多田作一郎著『AIの基礎を知る事典』(実務教育出版刊), 1997年。
- (4) 和多田作一郎著『新訂・ISDNの基礎を知る事典』(実務教育出版刊), 1995年。