

ライトサイジング環境でのシステム構築技術

川妻庸男

富士通

コンピュータ技術の発展とあいまって、ライトサイジング環境ではシステム化の対象は企業全体の共通業務ではなく部門の作業や業務となる。そのため、システム構築技術は、これまで以上に様々なニーズに対応しなくてはならない。今後は、個々のツールに依存した従来の考え方から、データの流れに着目して情報システムを整理し、そしてパターン化する方法に基づく新しいシステム構築技術が有効となってくる。この新しいシステム構築技術に基づいて、現在13個の基本的なシステムのパターンを得ている。そして、この基本パターンを約300種の商談で検証した結果、良好な成果をあげている。

System Construction Technology in Rightsizing Environment

TUNBO KAWATUMA

FUJITSU LIMITED

In the Rightsizing environment, with the help of the latest computer technology, target of the systematization is going to be the departmental operation and process, shifting from the corporate wide application. In this sense system tailoring technique must respond to the variety of this increasing needs. A new type of technique which analyses the system and clasifies it into a pattern focusing on the flow of data becomes effective instead of conventional technique which relies on the availability of the tools. The pattern of 13 basic systems was produced based on this new system tailoring technique. This pattern was applied to the 300 real business cases and good result was obtained so far.

1. まえがき

近年、高性能なUNIXマシンやパソコンの登場により、ライトサイジングをキーワードにしたシステム化が活発になってきている。それらが目指すものは従来ホスト集中のみを前提に開発運用されてきた情報システムの適正化であり、しばしば複数のCPUによる分散化を伴う。対象とするアプリケーションも、いわゆる基幹業務から部門業務まで広範囲になってきており、短期に開発する技術も要求されてきている。さらにオープンな環境ゆえに、それを実現するツールの選択肢も非常に増加してきており、従来のようにハードとバインドされた選択をする必要も無くなってきている。

ここで問題となるのは、従来の設計技術が一般にICPU内で処理する事を前提としており、かつ現実的には利用するツールにかなりのところ依存していることである。事実筆者らは、業務遂行上多数の情報システムのインフラを設計する立場にあるが、いままでの考えかたではなかなか前述のような環境には対応できないことが明らかになってきている。

そこで本稿では、ライトサイジング時代のシステム構築の特長と、それに対応するために筆者らが取り組んでいる、データの流れを中心としたアプローチについて述べる。合わせて、その成果の一つであるソリューションリストについても言及する。

2. 対象アプリケーションの変化

これまで多数の企業が開発してきたコンピュータシステムは、経理・人事・販売・物流・生産など、いわゆる基幹系システムと呼ばれるものである。これらは一般に汎用機やオフィスプロセッサを中核として実現されている。このようなコンピュータシステムの特徴として、

- 1) 企業活動のルールをシステム化している
 - 2) 多数の部門にまたがっている
 - 3) 共通性の高い業務である
- といった点が掲げられる。

こうした基幹系システムによって、現場では、全社的な企業活動に費やす時間や手間を大幅に削減できるよう

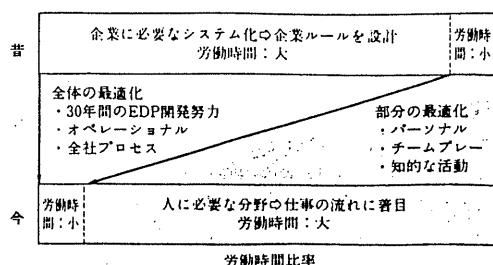


図-1 全体の最適化から部分の最適化へ

になっている。

ところが、このことが、逆に現場に固有な仕事に対するコンピュータシステムの貢献度を相対的に下げるといふ、皮肉な現状を生んでいる。そのために、企業全体ではなく、こうした現場での自分達に役に立つシステムが求められている。これが、これからのシステム開発の大きな対象となっている。

受注業務を行う一般的な基幹システムを考えてみる。

例えば、営業マンが受注を端末からエントリすれば、生産・梱包・出荷・請求といった一連の業務への伝達と調整をコンピュータシステムが行ってくれる。営業マン自身が電話で多数の関連部署と調整したり、複数の伝票を起こしたりしなくても、ビジネスそのものをより効率的に遂行できるようになったのである。つまり、図-1に示すように、昔は、企業のルールに大半の時間を割かなくてはならなかったのが、現在では僅かの時間で済むようになったのである。

その一方、顧客への情報提供などのサービスや顧客情報の管理など、受注に至るまでの様々な日常的な仕事や作業は、依然として営業の現場の人間の働きに任されている。そして、こうした活動の成果である受注エントリの瞬間でしか、利用者はコンピュータシステムとかわかっていない。つまり、利用者の目には、全体の最適化を目指したコンピュータシステムは、自分達が抱える大半の仕事には役立っていないと映るのである。企業のビジネスの拡大と、そのサイクルの高速化に伴って、こうした現場での活動は、より高度で複雑になっており、利用者の負担は増大する一方である。

ここに新たなシステム開発のニーズが出てくる。今後のシステム開発の対象として大きな比重を占めるのは、販売部門・製造部門・物流部門・経理部門・人事部門など個々の部門内の仕事なのである。ここでは、自分達に役に立つシステムが求められる。例えば、個々の営業マンにとっては、会社全体の顧客ではなく、自分が担当している顧客の情報を提供してくれるシステムであり、生産のスタッフにとっては、会社で製造している全製品ではなく、自分達が担当している製品の情報を管理してくれるシステムが求められているのである。(図-2)

もちろん、企業活動の一環である以上、当然企業全体をまかっている基幹系システムとのデータの連続性をも求められている。

3. ライトサイジング時代の課題

こうした新しいコンピュータシステムへのニーズに対する解として、システムの部品レベルで期待されるのがUNIX等に代表されるオープンな素材である。

低価格化、高性能化、小型化が進んで、部門の予算範囲内で、相当の機能や性能を備えたハードウェア製品や

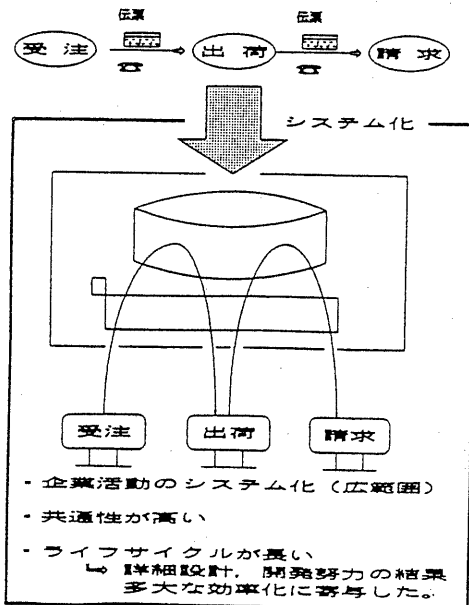
ソフトウェア製品を購入できるようになった。自分にベストフィットした規模の道具が手に入った訳である。そして、TCP/IPやSQ標準化によって、一つ一つの部品を相互に最低限の規約を確保できるようになっている。全体を何らかの形で協調させる拠り所が用意で訳である。それを利用し、個々の部門や作業に適な環境を作り上げることがライトサイジング考えている。

しかしながらこのオープン性を旨く利用するシステム構築上いくつかの課題が発生している。

従来の基幹系システムの開発では、開発者はの目的を実現するために、利用者のニーズを平準化してコンピュータシステムを設計し開発して、利用者はその標準化されたシステムを使わなかった。そのため、個々の利用者の作業に妙な仕様が現れることになる。例えば、数十種た2桁の数字で表せる種類の製品しか担当して門でも、全社共通の8~10桁もの標準化されたを入力しなくてはならないのである。

これに対して、個々の部門での作業を最適化コンピュータシステムでは、ある部署での実績

【企業の販売活動のシステム化】



【実際の受注活動とEDPシステムのサポート範囲】

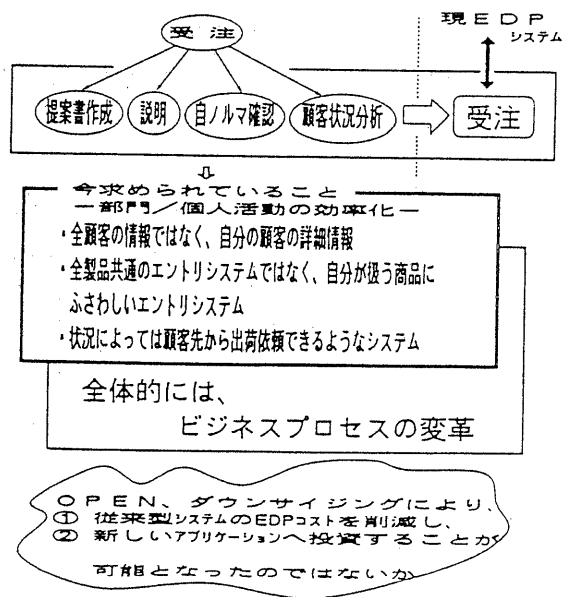


図-2 現状のEDPシステムのとらえ方

は1-2-3を、別の部署ではdBASE IVでのアプリケーションを使うといったことが起こる。設計者は、こうした多様な製品群を使い、各部門や個人に最適化したコンピュータシステムを設計し開発しなくてはならない。しかも、これらの製品群は多数のメーカーから提供されており、表計算ソフトウェア一つとっても、それぞれに特徴がある。ちなみに代表的な構成である、パソコンの表ソフトからUNIXのリレーショナルデータベースへのリモートアクセスを考えてみると、その実現方法は簡単に数百通りにもなることがわかる。(図-3)

このように、システム化のニーズが部門内で多様化し使うハードウェア製品やソフトウェア製品も多様化した時代に、どのようにして短期間でシステムの中身を決めて、アプリケーションを実現していくのが重要な課題となっている。

つまり、最適な組合せを選択する手段と、最適な組合せを短期間で実現する手段を確立する新たなシステム統合技術が必要なのである。

4. 問題解決に向けた着眼点

前述の課題を解決するためにわれわれが掲げた仮説は「数種類の基本パターンとそのバリエーションで組合せを表現できないか」であった。そして、その基本パ

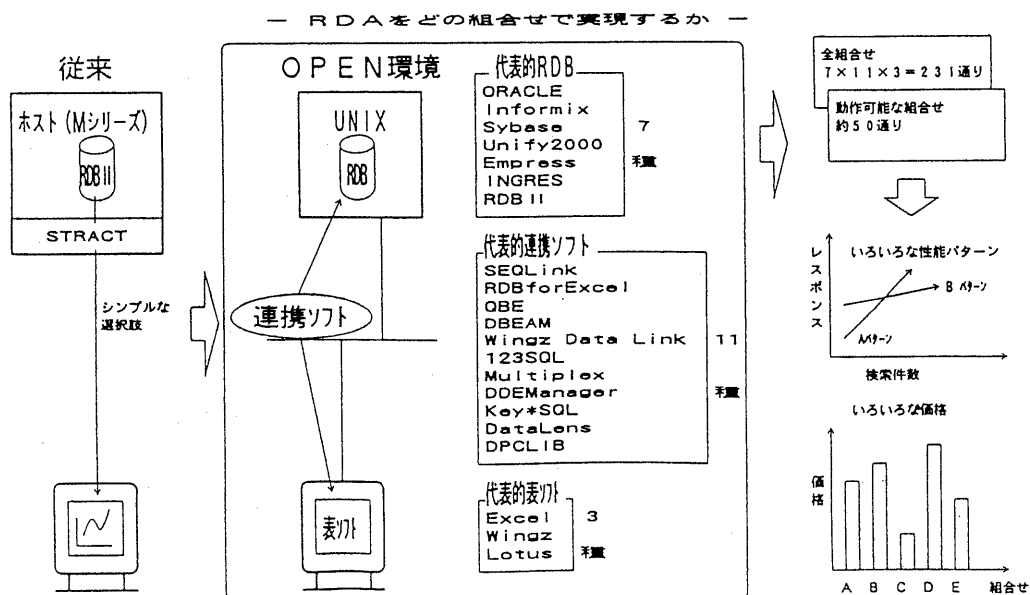
ターンを追求するために、データの流れに着目した。

その第一の理由は、過去450件にのぼるMESSAGE 90s商談対応の経験である。多くの対応を重ねるうちに、「異なる業務でも同じデータの流れになるシステムが存在する」ことに気付いた。例えば、社内教育の募集をしたり、有給休暇の取得計画を収集したり、販売予算と実績を管理したり、全社の予算編成を行ったりなどは、業務としては全く異なるものである。しかし、「管理元が雛形シートを配布し、各部署がそれに記入して管理元に返す」というデータの流れは共通である。

このような事例が多数現れたことから、データの流れに着目することで、基本パターンのための重要な鍵を得られると考えたのである。

第二の理由は、ハードウェアやオペレーティングシステム、サブシステム、汎用アプリケーションといった製品群と独立したパターン化を行えると考えたからである。

これまでのコンピュータシステム開発では、ハードウェア製品やソフトウェア製品が持っている、諸元・仕様・機能・性能を前提として実現方法を議論してきた。それは、そうしたものの選択肢が限定されていたからである。しかし、今回対象としているのは、いわゆるオープンと呼ばれる環境であり、そこにはハードウェア製品やソフトウェア製品の選択肢が多数存在する。そして、そ



の一つ一つの製品が変化したり、別の製品に容易に入れ替わったりする世界である。そのために、コンピュータシステムのインフラストラクチャに依存しないデータの流れに注目する必要があったのである。

5. データの流れに着目した問題解決の試み

まず情報システムの中身を業務層、方式層、インフラ層の3つの階層に分けて考えることとした。(図-4)

業務層に対する要件は、ツール無依存の業務分析・設計技術で、AA (Application Architecture) として従来から取り組んでおり、実績をあげつつある。

また、インフラ層の要件は多様なハード、ソフトの選択を適正に行う技術であり、情報の蓄積と選択の視点を充実させることであり、これも従来より取り組んでいる。

従来と違うのは、方式層である。従来は前述のように1CPU内で集中処理することが基本であったため、重視される必要はなかったが、分散化に対応するためにはどのように情報が流れるのかを重視する必要が出てくる。

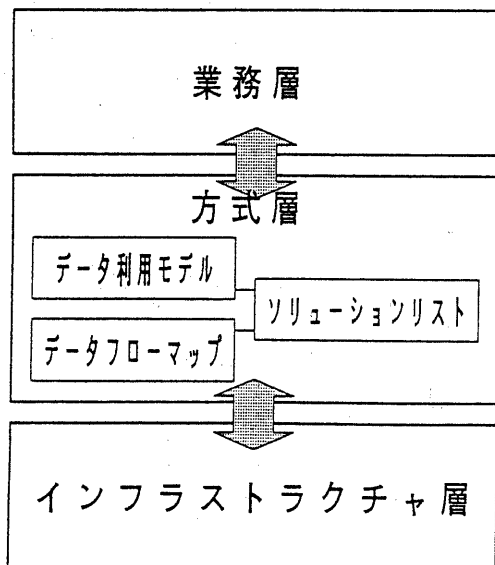


図-4 システム構築のレイヤモデル

筆者らはこの内容をさらに分析するためのフレームとして

- 1) データ利用モデル
- 2) データフローマップ
- 3) ソリューションリスト

の三つの体系を設定した。

そして、データの流れに着目して、それぞれの内容を検討した。この際、各体系でのデータの流れの視点を統一するために、四つの基本的なデータの流れを決めた(図-5)

- 1) コミュニケーション (人から人)
- 2) 情報活用 (コンピュータから人)
- 3) データエントリ (人からコンピュータ)
- 4) 環境整備 (コンピュータからコンピュータ)

ここでは、1)～3)に注目している。

5.1 データ利用モデル

データ利用モデルは、1991年5月21日に富士通

to \ from	人	コンピュータ
人	コミュニケーション	データエントリ
コンピュータ	情報活用	環境整備

図-5 基本的なデータの流れ

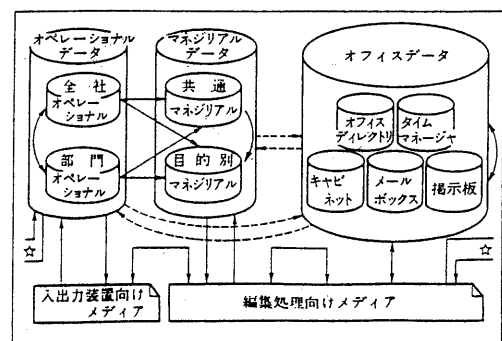


図-6 データ利用モデル

が発表したMESSAGE 90sの基本技術に位置付くものである(図-6)。

データ利用モデルは、業務やアプリケーションや利用者から見た、データの利用の仕方に基づいて、データの流れを記述している。したがって、コンピュータシステムのインフラストラクチャに依存しないデータの流れをパターン化している。MESSAGE 90s発表当初は、メインフレーム・オフィスプロセッサとパソコン・ワークステーションによる2階層のシステムの表現に利用していた。さらに、1992年9月5日のMESSAGE 90s強化発表時には、メインフレーム・オフィスプロセッサとサーバとパソコン・ワークステーションという3階層のシステムを表現するために、データ利用モデルをそのまま利用している。もちろん、単一のコンピュータという1階層のシステムも表現できる。

このように、データ利用モデルは、コンピュータシステムの形態やインフラストラクチャに依存することなく、様々なシステムのデータの流れをそのまま表現できるフレームになっている。

5.2 データフローマップ

データフローマップは、コンピュータシステムの構成上、論理的に可能となるデータの流れを把握するために作ったフレームである(図-7)。

多数のデータフローのパターンを分析するために表記法にも工夫をしている。データフローマップを分析することによって、

- 1) 現時点で意味のあるデータの流れのパターン
- 2) 現在の技術で実現されていない有益なデータの流れ

のパターン

を得ることができた。

3階層のCPU構成におけるデータフローは、筆者らの検討は約3万通りにもものぼったが、実際のシステムの上で事例として現れてきたものは数10通りでは説明できた。それを更にツールとして使いやすくするために絞りこんだのがソリューションリストである。

5.3 ソリューションリスト

データフローマップから得られた基本パターンを現実のシステム開発で利用できるように表現するために、つぎの二つの視点を定めた。

第一の視点は、データ利用モデルから得られるデータの利用の仕方である。これを以下のように整理した。

1) コミュニケーション

統合OAや電子メールを共通のインフラストラクチャにする展開である。人と人とのコミュニケーション、掲示板、情報交換、電子メール、電子キャビネットといった使い方が該当する。

2) 情報活用(揭示型・検索型)

蓄積された情報を有効に活用したいというニーズは、「定型情報を再加工したい」というものと「自由に検索・分析したい」という二つに分類できる。

前者を「状況を確認でき、そのまま情報を再利用」という視点で「揭示型」と呼ぶ。例えば、紙の帳票を表計算ソフトウェアのデータとして配布することなどが該当する。後者を「利用者が試行錯誤を行いながら想定して

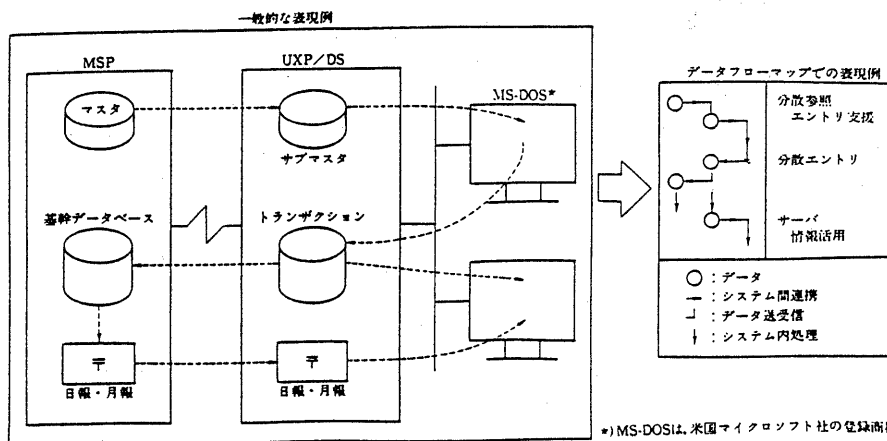


図-7 データフローマップの例

いる仮説を検証していく」という視点で「検索型」と呼ぶ。例えば、表計算ソフトウェアから、直接ホストやサーバのデータベースを検索し、利用することなどが該当する。

3) セルフコンピューティング (表型・DB型)

利用者主体の部門システムを実現するという視点に立つとき、エンドユーザコンピューティングという用語が持つ曖昧さは混乱をもたらす。そこで、利用者自身が情報を活用することを前提に、データエントリまで行うことを筆者らはセルフコンピューティングと呼んでいる。セルフコンピューティングは、つぎの二通りの使い方に分類してある。一つ目は、表形式で一括入力を行う「表型」と呼ぶ使い方である。例えば、表計算ソフトウェアに受注データを一括して入力し、表データとしてホストやサーバに送付することなどが該当する。二つ目は、データベースに対して項目単位に入力する「DB型」と呼ぶ使い方である。例えば、データベースソフトウェアや電子伝票機能を用いて、オーダごとにデータをホストやサーバに格納することなどが該当する。

第二の視点は、利用者から見たコンピュータシステム

を構成するハードウェアやソフトウェアの配置である。これを以下のように整理した。

1) ホスト・パソコン (2形態)

ホスト集中型のシステム構成である。

i) Mシリーズ・パソコン (M*と表記)

ii) Kシリーズ・パソコン (K*と表記)

の二つの形態とした。

2) ホスト・サーバ・パソコン (5形態)

ホストとパソコンの間にサーバを挿入した、いわゆる分散を意図したシステム構成である。

i) Mシリーズ-Kシリーズ・パソコン (MK*と表記)

ii) Mシリーズ-UNIXサーバ・パソコン (MU*と表記)

iii) Mシリーズ-MS[®]OS/2サーバ・パソコン (MO*と表記)

iv) Mシリーズ-NetWareサーバ・パソコン (MN*と表記)

v) 他社ホスト-UNIXサーバ・パソコン (XU*と表記)

システム構成		ホスト・パソコン		ホスト・サーバ・パソコン					サーバ・パソコン		
システムスタイル		M-PC	K-PC	M-K-PC	M-UN-PC	M-OS2-PC	M-NW-PC	他-UN-PC	UNIX-PC	OS2-PC	NW-PC
コミュニケーション		M1 							U1 		
		オフィス連携パターン									
情報活用	指示	M2 							MN2 		
	検索	M3 	K3 		MU3 	MO3 			XU3 		
セルフコンピューティング	表	M4 									
	DB				MU5 				MN5 	XU5 	

M : M シリーズ
NW : NetWare サーバ

K : K シリーズ
PC : パソコン

UN : UNIX サーバ
他 : 他社ホスト機

OS2 : MS[®]OS/2 サーバ

図-8 ソリューションリストの一覧 (パソコン連携)

- ・現状のメインフレーム環境に影響を与えず情報活用を推進
- ・部門や地域の特性に合わせた情報活用を実現

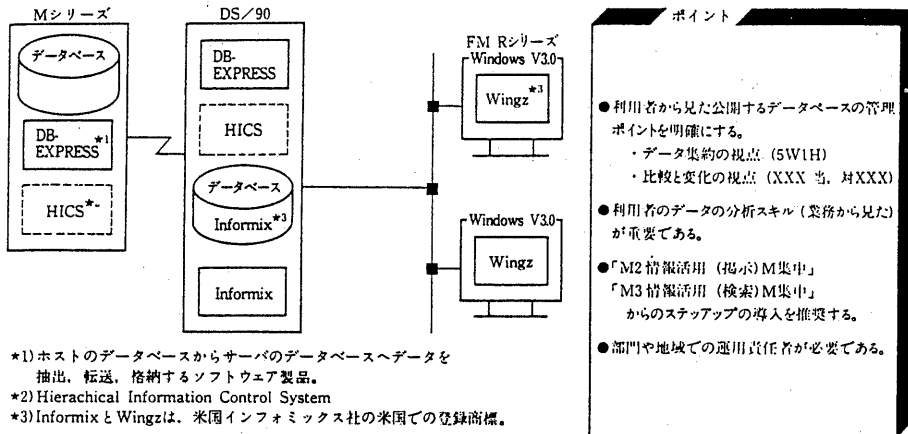


図-9 ソリューションリストの例-情報活用 (検索) UNIX 分散

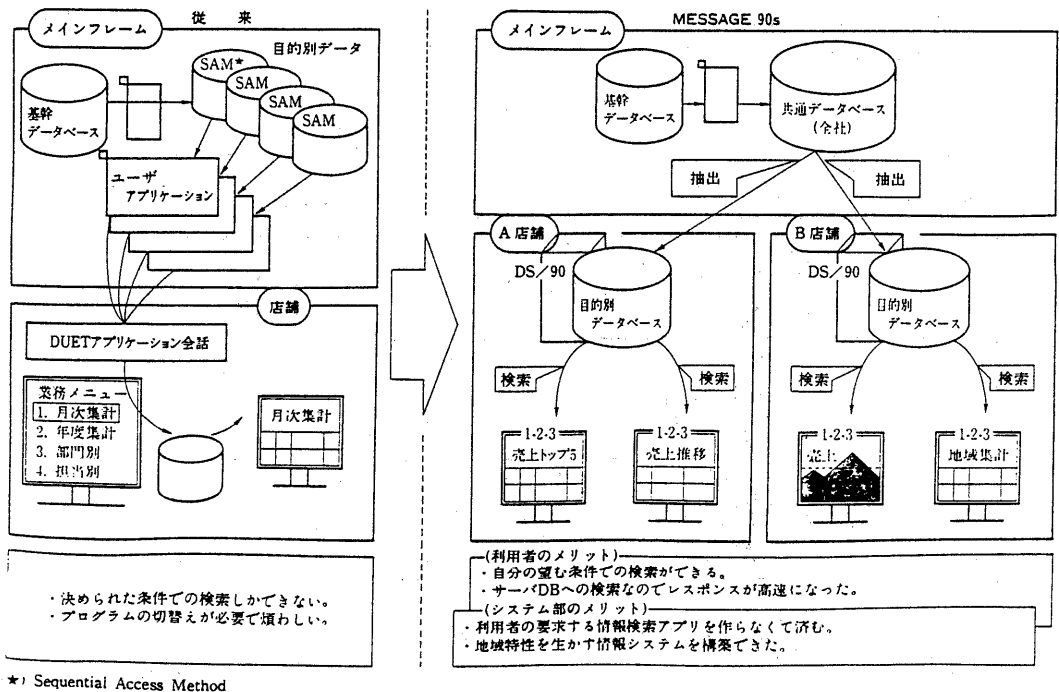


図-10 ソリューションリストによる事例-地域別店舗別での売上分析

の五つの形態とした。

3) サーバーパソコン (3 形態)

部門に閉じた、独立システムの構成である。

i) UNIXサーバーパソコン (U*と表記)

ii) MS OS/2サーバーパソコン (O*と表記)

iii) NetWareサーバーパソコン (N*と表記)

の三つの形態とした。

これら二つの視点に基づいて、縦5行×横10列の計50個のセルをもつフレームを設定する(図-8)。このようにすると、一つのセルの中に、前述のデータフローマップで得た一つの基本パターンが対応することになる。

ここでのわれわれの方針は、この50個すべてを基本パターンとするのではない。できるだけ単純な基本パターンを設定し、他をバリエーションとして対応することである。そこで、過去の約300件の商談対応(92年8月現在、MESSAGE 90s推進課対応分)の経験に基づき、最終的に13個に推奨する基本パターンを絞り込んだ。これが、ソリューションリストである。

図-9、図-10は、検索型の情報活用を、Mシリーズ-UNIXサーバーパソコンという配置で実現するソリューションリストの例である。図-9は、概要構成図である。この図によって、業務に必要な要件を大まかに整理でき、必要な部品の配置を知ることができる。実際の資料には、さらに詳細な情報を添付している。図-10は、このソリューションリストで実現したシステムの事例である。

これらが、基本パターンであり、個々の製品を別の製品に入れ換えていくことで、バリエーションを作りだせる訳である。

6 効果

6.1 ソリューションリストの有効性

ソリューションリストを発表当初、われわれは、商談のヒット率は70%程度であると予測していた。しかし、その後の新たな約150件の商談対応(93年4月現在、MESSAGE 90s推進課対応分)で適用すると、ヒット率が90%を越えた。その内容を分析すると

- 1) 複数の商談に対して、同一のソリューションリスト

で対応できる。

- 2) 商談の要件によって製品を入れ換える場合でも、必要となる技術的機能が明確になっており、検証が容易である。

- 3) 要件が曖昧な初期段階の提案でも、結果的にニーズにフィットする。

- 4) ソリューションリストごとに作成したデモンストレーションシステムが、ニーズを先取りしている。

というソリューションリストの特性が分かった。

このように、ソリューションリストは予想を上回る高いヒット率を示している。これは、われわれのアプローチ方法が、かなりの確率で正しいと考えられる。

6.2 インフラ設計への有効性

データフローは業務処理とインフラを接続する情報である。集中処理時代にはアプリケーション設計の中に埋没していた情報を、外に取り出したとも考えられる。従ってこのデータフローの代表的なパターンの蓄積は、業務を詳しく分析しなくても、ポイントを抑えれば適用するインフラの構成を教えてくれることになる。

筆者らの経験では、代表的なデータフローを記述したワークシートがインフラの概略を設計する上で有効であると考えられる。

7. むすび

われわれは、ダウンサイジング・オープン化時代におけるシステムインテグレーションのための基礎技術と、システム構築手法を確立することを目的に、データ利用モデル、ソリューションリスト、データフローマップを開発した。

このうち、ソリューションリストは、ようやくビジネスの中で検証し始めたところである。もちろん、商談に対するヒット率は、現時点では期待以上である。しかし、データの流れは、本来、業務の流れに依存するはずである。したがって、業務分析の結果の一部がデータフロー表現になるようにすることが今後の重要なテーマであると認識しており、ライトサイジング時代の有効な手法として確立してゆきたい。