

California.

- 3) The Network Strategy Report: Forester Research, Inc. One Brattle Square, Cambridge, MA 02138.
- 4) The Dinosaur Myth: Why the Mainframe is the Cheapest Solution for the Most Organization: Xephon, PO Box 1059, Oviedo FL 3265.
- 5) Gartner Group/IRM Spring Conference: May 26&27, 1992, Gartner Group.

(平成5年7月2日受付)



中島 丈夫 (正会員)

1944年生. 1967年神戸大学電気工学科卒業. 1969年同大学院電気工学専攻修了. 旧姓深津. 1969年日本アイ・ビー・エム(株)入社. フィールドSEを経てシステム評価部門・大型システム担当. その後, 本社SE技術担当等を経て現在SE研究所所長. 電子情報通信学会会員.

ダウンサイジングとオープンシステム, 何がそうさせるのか?†

村井 修造††

1. 発端と先駆者

最近, 大型汎用計算機を取り巻く状況が慌ただしい. 昨今の“ダウンサイジング”ブームにみるごとく, これまで大型汎用計算機上で行われていた処理をより小さな, 異なった処理方式のシステム(主として“オープンシステム”と呼ばれている考え方や, “クライアント/サーバ”と呼ばれている処理方式)で置き換え可能なのではないかという議論が行われている.

こういう議論では当然「オープンシステムとは何か」ということも重要ではあるが, ここではとりあえず「固有のベンダから独立しており, 広く一般に利用されている標準規格に合致するように設計され, 具体化された製品や技術で, 可能なかぎりの接続性や互換性を確保しつつシステムを構築していくこと」という暫定的な定義の下に話を進めていくことにする.

さて, 今日のオープンシステム環境で使用される技術的要素の多くはすでに70年代に構想され, 基礎的な研究が行われていた. 80年になるとボストン郊外のアポロ・コンピュータ社(現, ヒューレット・パカード社ワークステーション事業部)が世界最初の商用ワークステーション“Apollo Domain DN100”を商品化した. このシステムは

基本的にはゼロックス社パロアルト研究所の研究成果を元にしていたものの, 非常に多くの点でそれまでのコンピュータとは異なっていた. その中でも最も特長的と思われる点は以下のとおりである.

1. 非対等通信/集中処理の発想から生まれたターミナルという概念がない.
2. システム資源として接続が前提となっている対等通信 LAN

1. は 2. の対等通信をベースにしているということと深く結び付いている. ターミナルという発想は基本的には非対等通信/集中処理の概念だからである. このようなユーザの本当にローカルで処理が行われるというのは一見当時のPCと同様の考え方ともとれるが, 2. のアプローチがあまりに衝撃的なためシステムとしてはまったく違ったものになっている. アポロのシステムではノード(ワークステーション)は徹頭徹尾, 対等通信のネットワーク(トークンリングの技術を採用)に接続されて使用されるのを前提としてハードウェア, OS, GUIなどのすべてが設計されていた. だからアポロのシステムではLANに接続されているというのが普通で, ノードがスタンドアロンで使用されている, というのは非常に特殊, かつ不完全な状態であると定義される.

このため, このシステムではスタンドアロンで使用する際にもノードが1台のみの疑似的なネットワークを設定し, 小さなループにしたLANに接続する必要があった. ここには今日のCoope-

† Downsizing by Open System, The Unknown Reasons by Shuzo MURAI (Senior Systems Consultant, PSO#5, Professional Services Organization, Yokogawa-Hewlett-Packard, Ltd.).

†† 横河ヒューレット・パカード(株)システムエンジニアリング本部 SE5部

ative Processing (相互協調処理—分散処理の完成された形) にまで通じる発想の逆転がみられる。

さらに、当時の技術資料やマニュアルを見るとこのシステムの意図したものは大型汎用計算機などの TSS 環境を分散処理で置き換えることであると明確に書かれている。

アポロのシステムは筆者の知るかぎり、大型汎用計算機を分散処理で置き換えようとした最初の製品であり主としてエンジニアリング分野ではあるが、実際に多くの現場で成功した。

アポロのシステムが提示した方向性は現在もオープンシステムをベースとした分散処理という形で追求されている。これを称して“ダウンサイジング”の先駆者と言うことはできると思うが、このような処理方式の考え方は 80 年当時、ボーイング社の IPAD (Integration Program for Aerospace Vehicle Design) プロジェクトにもみられたものである。

これらの先駆者たちの議論では、主として処理方式の変革が最も重要であっていわゆる“サイジング”についてはほとんど重要視されていない。

筆者の基本的な考え方も IPAD を知った当時から変わっていない。つまり、重要なのは処理方式／処理形態であって、“サイジング”はその結果でしかないということである。

2. 「ダウンサイジング」と言う表現

今日のダウンサイジングと呼ばれるようなシステムを構想していた人々が当時は必ずしもサイジングを重要視していなかったというのは暗示的である。そのころは 90 年代初頭のようなタイプの不況がやってくるとは考えられなかったのかも知れない。それに筆者の感じるころでは当時のエンジニアの技術的な検討は今日より政治的判断やセンセーショナリズムから自由であり、より理想的なシステムを考えることができたのではないだろうか？

「大型汎用計算機は生き残れるか？」というテーマを考えるうえでも“サイジング”という要素を重視し過ぎると本質を見失う。この議論はあくまでも処理方式／処理形態と企業の業務のあり方が見合うかどうかという点に本質があるように考えられる。もちろん今日においては情報処理コストの低減は一般的留意事項であるから、これも情報

システムを考えるうえでの一つの前提ではあるが経済性はむしろ情報システムの投資対効果という見地から考慮すべきであろう。

本質を覆い隠すという意味で、“ダウンサイジング”と言う表現はあまり良いものではない。この表現はどこかネガティブな響きがあり、システムを変えることでこれまでにない、より良い環境を実現するという言い方には聞こえない。それでもコストをセーブするという側面は非常によく分かるので、多くの場合企業のトップの方に対して、逆らい難い魅惑の響きをもって迫る。

こうした方々に筆者が必ず申し上げるのが、「ダウンサイジングは貧者の汎用機を目指してはいけない」ということである。これは言い方を変えると、重要なのはコンピュータのサイズをどうこうするというのではなく、集中処理を対等通信の分散処理に換えることに本質があるということである。

3. 技術的背景—集中処理と分散処理

現在のように大型汎用計算機をオープンシステムで置き換えられるのではないかと注目されるようになったきっかけには当然、UNIX マシンや PC などのコンポーネントの処理能力の向上があげられる。だから、CISC から RISC への CPU チップ・テクノロジーの変化がダウンサイジングの潮流を決定づけたのだ、という考え方も存在する。しかし処理能力その物について考えるとわれわれベンダも含めて、大型汎用計算機とは同じ基準で比較しているわけではないので、単純にその点を数値的に検討するのはおかしい。実際大型汎用計算機のスペックマーク値などというものは聞いたことがないし、たとえば企業の基幹業務系で行われているようなバッチ処理の性能を直接比較したような例はないわけではないが非常に少ない。筆者の周辺のエンジニアは比較的バッチ性能を重視しているほうだと思うが、業界全体の注目度は低い。重要なのは処理方式がそもそも違うのだから性能検討の基準も異なるのだということである。

大型汎用計算機の集中処理とオープンシステムの分散処理は技術的に対立する概念である。これはオープンシステムが、前章で言及した対等通信をそのインフラとして仮定するというだけではなく、情報処理における負荷の集中にどのように対

処していくかという方向性に大きな違いがあることによるものである。大型汎用計算機においては負荷の集中に対して、キャパシティの大きな機種を導入していくことでこれを吸収していく。

これに対して、オープンシステムによる分散処理環境では分散によって負荷を拡散する方向で吸収しようとする。この発想はクライアント/サーバ・モデルの原点とも言えるもので、実際にシステム化を進めていく際にも常に念頭に置く必要がある。というのは肝心の負荷分散についての検討が十分に行われず、クライアント/サーバのような形式のみが先行するケースが多々あるからである。最悪の場合はコンポーネントは UNIX マシンや LAN を使用して一見クライアント/サーバ型にみえるものの、実際の処理分割に対する検討はほとんど大型汎用計算機の発想で行われているという例がある。こういったアプローチではシステム化による性能向上は望めない。大型汎用計算機とオープンシステムは異なるものであり、どこが違うのかということをはっきりと認識したうえで、システム化を計画する必要がある。

4. 大型汎用計算機とオープンシステム、何が違うのか？

大型汎用計算機とオープンシステムの本質的な違いを認識することは重要である。この理解をユーザに徹底してこなかったのは筆者のようなベンダのエンジニアも含めて、業界全体の情けなさである。この認識なしに「ダウンサイジング」を行うことが、「貧者の汎用機」を生むのである。

大型汎用計算機は交通機関にたとえると、幹線鉄道のようなものであり、オープンシステムはそれに対して自動車のようなものであると言える(図-1)。この比較をもう少し論理的に言うと、

大型汎用計算機における量的/性能的検討は

情報システムを交通手段(メディア)であると考えたと

大型汎用計算機は幹線鉄道である。

- ※ それは大量のいろいろな荷物(データ)を運ぶ。
- ※ 大量のいろいろな乗客を乗せる。
- ※ 比較的安全で、日本においては時間も正確である。
- ※ 事故、故障の波及効果は甚大である。
- ※ 目的地はあらかじめ決まっている。
- ※ あらかじめ限定された目的地には早く着くかもしれない。
- ※ 新たな計画には国家予算の計上が必要だった。
- ※ しばしば経営的に赤字である。

常に、

- 決定論的であり、確定的である。

それに対してオープンシステムにおいては常に、

- 確率論的であり、統計的である。

とすることである。

大型汎用計算機の環境は比較的不確定な要素が混入しにくいものである。ターミナルのキャラクタ数は決まっており、そこから一回のトランザクション量も容易に計算可能である。通信の回線速度もだいたい決まっており、トラフィックの量も比較的正確に見積れる。さらに OLTP 環境のバッファ・サイズなども他の要素と密接に関連があり非常に計算しやすいものである。これに対してオープンシステムにおいては、しばしばマルチ・ベンダのシステムとなり、そもそも確率論的なメディアである LAN に UNIX のサーバがあり、PC があり、ワークステーションがあり、X-Window があり、Windows があり、それぞれがこと対象をネットワークに限定してもどのような影響を及ぼすものか、既存の手法では見積り難い。少なくとも大型汎用計算機で通常行われるような量的/性能の見積の基準をオープンシステムに導入するといつまで経っても検討が終わらない。実際このような異にはまって、膨大な時間を量的検討に費やしている例は多々ある。しかし、この点に言及して大型汎用計算機の環境のほうが優れていると短絡的に判断することには異論がある。なぜなら大型汎用計算機においてこのように正確な量的/性能的検討が可能であるということは「システムの制約/可能性の排除」と不可分の関係にあるからだ。

実際オープンシステムで情報システムの検討を進めていく際の困難さはそのシステムを使用することによってユーザの環境やビジネスがどのよう

それに対して、オープンシステムは自動車である。

- ※ 乗客(ユーザ)が自分の意志で運転できる。
- ※ 目的地は比較的自由に選択できる。
- ※ たくさん走ると渋滞する。交差点も渋滞の原因になる。
- ※ 運転は時として楽しく、それ自体目的になりうる。
- ※ 簡単に凶器になりうるため、免許が必要である。
- ※ 交通違反を取り締まるのは大変な作業である。
- ※ 頻繁にモデルチェンジが起こり、不必要に速くなる。
- ※ 改造車両が群れをなして車線を逆行することもある。

サーバもしょせん自動車、バスか、トラックでしかない。

図-1

に変化するかという内容を含んだ茫漠とした可能性が背後にあるために起こる。量的／性能的検討だけではなくこれまでのオープンシステム導入における最大の問題はその世界が茫漠としたものであるにもかかわらず、有効な道標、「ロードマップ」が存在していなかったことである。初期のダウンサイジング／オープンシステム導入事例にメンテナンス・コストが異常に増大している例やパフォーマンス的に苦勞している例、カット・オーバー（システム開発が終わり、サービス開始となる時）の瞬間に陳腐化するほど開発期間がかかっている例（もっとも今日では開発に2年もかけるとカット・オーバー以前に陳腐化する）などがみられるのは茫漠とした世界を手探りで進んでいるからである。さらに最悪のケースはコンポーネントのすべてをオープンシステムで使用されるものを採用したにもかかわらず、大型汎用計算機の開発メソッドを導入して大規模開発を行った例である。この場合ハードウェアを除く導入コストが大型汎用計算機の場合を上回るということも実際に起こる。筆者の知るかぎりにおいて今日オープンシステム導入における“ヒドゥン・コスト”（システム開発／導入時に明確化できないと考えられているが確実に発生するコスト）と呼ばれているものは、このような“違いを認識していない”状況から生みだされたものが多い。

5. ビジネス・プロセス・リエンジニアリング

米国などの、ダウンサイジング関係のコンフェレンスには、“ビジネス・プロセス・リエンジニアリング”のコンフェレンスも併設される例が多い。

“ビジネス・プロセス・リエンジニアリング”とは「業務に対して根本的な再検討を行い、組織構造やビジネス上の処理、個々の作業のやり方などを再検討すること」を指すが、このような検討がなぜ情報システムの“ダウンサイジング”と併せて行われるかと言うと、今日の企業活動の基盤が変化しつつあることに対応し、情報システムも変わっていくと考えるからである。

過去において企業活動の鍵となっていた要素は量的な意味でも質的な意味でも「労働力」であった。だから多くの企業にとって最初の情報システム構築は基幹業務系から行われた。基幹業務系

システムの獲得目標は定型化された業務を無人化／自動化することによる省力化であったからだ。

ところがリエンジニアリングを指向するコンサルタントの多くは今日、企業活動にとって鍵となる要素は「労働力」から「情報」に変わったと考える。このことが昨今の情報系システムのオープンシステム化／エンドユーザ・コンピューティング化の一要因であるというのだ。

情報系システム（意思決定支援系システム—戦略的アプリケーションと呼ぶ人もいる—）においてはもはや議論の余地はないと思う。大型汎用計算機が適しているのか、オープンシステムが適しているのか、すでに勝負はついている。ほとんどの場合について今日のニーズにフィットするのはオープンシステムによる構築であると言っていいと思う。問題は現在大型汎用計算機で構築されている基幹業務系システムにある。

ラディカルなコンサルタントはこれまで大型汎用計算機で構築されてきた基幹業務系分野にも再検討が必要であると言う。なぜなら大型汎用計算機による基幹業務系システムは大きく、複雑になり過ぎている。しかもIT（情報処理技術）が固有のベンダに依存しているため小回りが利かず、必要なシステム変更がタイムリにできなくなっている。ましてや新規開発など現在のマーケットのスピードを考えるとまったく現実的ではない。先進的なCIO（Chief Information Officer—情報システム部門のトップで情報システム戦略を策定する立場にある役職）の方の中には、これを称して「大型汎用計算機のスケール・ディメリットが出ている状態」と呼ぶ人もいる。

今日多くの企業で、業務系システムはブラックボックス化が進んでいる。非常に大規模な開発が行われたにもかかわらず、その処理の全貌を把握しているのはごく一部の担当者である。実際の業務を行っている部署からみると、「よくわからないが、凄い量のデータが扱われて処理されているらしい」となる。たとえば人事のシステムでも、人事部担当者がコンピュータ・システムの中で何が行われているのか把握していないということも多い。

企業がこのような状態にあることを（実際は多くの企業がそれに近い状態にある）リエンジニアリングの立場では危険な状態と考える。これは情

報システムが企業の中で無意識化している状態と言える。情報が企業活動の「鍵」となる要素なのであれば、精神分析にたとえて言うと、無意識を意識化するプロセスが必要であり、「情報」は最大限活用されなければならない。このことが、これまで大型汎用計算機で行われてきた基幹業務系でもオープンシステムの導入を検討する理由の一つとなっている。なぜならば、もともとオープンシステムは「いかにして効率的に情報を利用するか」ということを優先させて考えるシステムに適したものだからである。たとえ現状のオープンシステムが現行大型汎用計算機に比べて、管理運用の見地から比較的脆弱にみえるものだとしてもである。

6. 何が起こるのか？

基本的にオープンシステムは「情報を活用する」とか「使いやすい」というところに美点がある。これに対して大型汎用計算機は「情報を守る」とか「管理する」という業務に向いているようにみえる。だから筆者はこれまで、両者の特長を活かしたシステム構築が今後も行われると考えてきた。

しかし、ここ約一年間日本でもトップクラスの企業のシステム関係者にお話を伺う機会があったが、大型汎用計算機による基幹業務系システムの周辺にとんでもない状況が起きつつあるということに気がついた。それは人材の問題から既存業務系システムの崩壊が始まるのではないかということである。それも想像以上に早い時期に。大型汎用計算機の環境は開発・導入、運用など、すべてにオープンシステムからみると人海とも言うべき人的資源を必要とする。ところが、今日のようなマーケットの状態ではこの「人海」を供給してきた周辺ソフトウェアハウスも大型汎用計算機人材の育成にこれまでのように投資するわけにはいかない。たとえば、大型汎用計算機のソフト開発ピラミッドの最下層に位置するコーダ（主として単純なプログラミング作業に従事し、設計を行わないレベルのプログラマ）を大量に抱えたソフトウェアハウスの経営状態が悪化していることはよく知られている。

メインフレーム・ベンダや、ベンダ系のソフトウェアハウスにおいても人材育成はオープンシ

ステムにシフトしつつあると聞く。このことが原因かどうかはわからないが、大型汎用計算機による基幹業務系システムの周辺では担当者の高齢化が進み、どんどん層が薄くなりつつある。筆者のようなオープンシステムのエンジニアやベンダは大型汎用計算機のことを批判的に「Legacy System」と呼ぶが、まさに「遺物と化して受け継がれているシステム」である。本当に既存のシステムを「伝承」し、新規機能を追加し、維持していくことが可能なのだろうか？

現在の状況は最後の大規模開発が終わった状態と言えらると思う。金融機関の第3次オンラインのように1,500億円も大型汎用計算機による基幹業務系システムに投資するようなことは二度とできないだろう。投資するお金がないのではというはなく、もはや「人海」がないのである。

80年代に叫ばれた「ソフトウェア・クライシス」は思わぬところから現実となりつつある。

7. 何をなすべきか？

つまるところ、今日起こりつつあり、またそれに対してなすべきことは情報処理全般の質的変革であり、量的な変化は結果として生じるものであるということを述べてきた。

ここで、最後に三点ばかり重要な課題をあげたい。

まず、オープンシステムを標榜しているベンダはそのマーケットに対して、ソフトウェアハウス、ユーザとともにオープンシステムによる基幹業務系システムの具体的なイメージを明確に提示できなければならない。また、それはどのようなツールを用いて行うか？ といった議論だけではまったく不十分であり、リエンジニアリング的に業務や組織のあり方まで再検討を行ったものではなくてはならないだろう。大規模な業務系システムには公共性の高いものもあるわけで、現状の大型汎用計算機による基幹業務系システムの現状を考えると今後1～2年以内に再検討が行われ、新しいシステムの具体的なイメージが出てくようにならないと、今後サービス性の低下や変化した業務にシステムが追従しきれない可能性も出てくると考えられ、危険である。

オープンシステムによるシステム構築で、量的／性能的検討が難しく運用管理の実体が分かり

にくいということ述べた。この問題についての現状の解決策は「パイロット・システム」の構築である。今日ソフトウェア開発において「プロトタイピング」は大型汎用計算機上でもすでに行われており、オープンシステムやコンポーネント上では常識化している。これは一般には仕様の確認や、ニーズの抽出に利用されているが、「パイロット」とはプロトタイピングのアプローチを運用/管理を含めたところまで拡張したものと言える。このアプローチによって、実際のシステム上で発生する量的な問題もより良く把握できるようになり、ユーザや業務の実体に適した運用管理の計画も立てられるようになる。筆者らオープンシステムのエンジニアからみると、「パイロット」での検証フェーズなしにシステムをデリバ（システムの引き渡し/実運用の開始）すること自体、まったく考えられない。これはソフトウェアの開発ライフサイクルを越えたメソドロジの拡張である。たとえば、現在 HP には「オープンシステム・ロードマップ」と呼ばれるシステム導入のプロセス定義と対応するサービス体系があるが、この中でも「パイロット」は重要な位置を占める概念である。

最後に人材再教育の問題がある。今日、大型汎用計算機の周辺人材に層の薄さが目立つと述べた。これは事実だと思う。しかし、かつて確かに「人海」が存在したのも事実である。彼ら/彼女らはどこへ行くのだろうか。ほとんど事務職と化していた「コード」と呼ばれるエンジニア人口は膨大であった。オープンシステムというマーケットがあるのなら、その業界全体で対策を考える責任があるはずである。そして事実こういったニーズは大きい。たとえば、YHP のカスタマ教育部門では今年になってから「コボル・プログラマのためのC言語入門」というコースを日本独自に開設したが、大好評となっている。

先日、筆者はあるパネルディスカッションの席上、パネラとして「基幹業務系人材崩壊説」を問題提起した。会場には国産のメインフレーム・ベンダの方も多数おられたようである。猛烈な反発を食らうことを覚悟して身構えていたが、意に反しておおむね納得されてしまい、拍子抜けしてしまった。筆者のようなものが口に出して発言しても、それは言ったかどうかの問題だけで、現実の

ほうはすでに後戻りできないところまできているようである。

参考文献

発端と先駆者

初期のワークステーションのコンセプトについて

1. DOMAIN システム概要書

DOMAIN ARCHITECTURE

A Technical Overview

横河・ヒューレット・パッカード株式会社 1989

PART No. S09-02518 J

初期のアポロ・ドメイン関係の資料は現在入手困難である。

上記資料はあくまで概要ではあるが、その設計思想について触れている。

ボーイング社の IPAD プロジェクトについて

1. TECHNICAL PRODUCTIVITY

The Central Issue Of Human Problem Solving

Ralph E. Miller, Jr. and Herschel G. Owens

Presented at AIAA International Annual Meeting

May 25-27, 1982

Baltimore, Maryland

2. HUMAN PROBLEM SOLVING AND ITS RELATIONSHIP TO COMPUTER AIDED ENGINEERING SYSTEM

Ralph E. Miller, Jr.

Presented at 日本エンジニアリング振興協会(ENNA)

October 1982

IPAD プロジェクトは 1972 年に NASA からボーイング社に移管された宇宙船設計のための情報処理インフラストラクチャ開発プロジェクトである。

上記の資料は残念ながら現在入手困難であると思われる。

何が起こるのか？

ソフトウェア業界の変化については多数の著作、雑誌などに取り上げられているが、オープンシステムの動向との関連で、下記の文献をあげたい。

1. コンピュータ業界・ソフト業界にいま何が起きているのか

ソフト企業経営研究所代表 内海一郎氏

2. 公認会計士がみたダウンサイジングの実実際

中央新光監査法人公認会計士 久保光雄氏

Open System Review

1993 年 6 月号

Unix Business Association 刊

(平成 5 年 7 月 2 日受付)



村井 修造

1955 年青森市生。法政大学文学

部史学科卒業。専攻は現代思想史。

卒業後、数年間音楽雑誌のライター

を経験。1982 年ソフトウェア・ハウ

ス入社。1988 年当時の日本アポロ・

コンピュータ社に入社。マーケティ

ング SE として金融分野を担当。翌年同社のヒューレット・パッカード社への吸収合併により横河・ヒューレット・パッカード社社員となり、今日に至る。現在はシステムエンジニアリング本部所属のコンサルタントとして金融、流通、製造等、主としてビジネス分野のシステム構築、ダウンサイジングを担当する。