

## NUANCE の 思想

### The Basic Concept of *NUANCE*

平 岡 信 之

Nobuyuki Hiraoka

#### 1. はじめに

本学では、この94年度に基礎教育用計算機設備の更新が行われた。当年度夏期休暇中にその更新に関する主な工事が行われ、後期より新規設備に基づく授業が行われている<sup>1)</sup>。本稿はその新規システムに関して、いわば思想的な解説を試みるものである。

筆者は、92、93年度計算機システム委員会の一員として、この設備更新の準備段階から関っており、旧設備の窮状の上申に始まり、本年度初頭に行われた担当者選定のための要求仕様書<sup>[1]</sup>

(以下仕様書と呼ぶ)の原案作成や、それに先立つ技術動向調査等の作業に主動部として直接タッチしてきた。本稿は主としてその作業に関するものである。一言で言うと筆者は新規システムの基本設計(或は概念設計と言うべきか)を行なったことになる。この作業は、話の流れからして、学内の雑務の1つに位置づけられるものである。筆者自身も当初は自分の研究テーマとする気持ちもなく、現状の問題を少しでも解決するために仕方なく買っただけのものであった。が、やがてこの問題は、様々な制約のもとにシステムを構築するという、難しい(ある意味では面白い)問題と化し、結果的には、情報システムの専門家のはしくれである筆者が、その研究時間と専門知識のある期間総動員することを必要とし、筆者にとってはその過程で情報システム或は教育用システムのありかたについて調査及び考察することも含め

1) その構築作業(主としてソフトウェアの整備に関する残務)は、本稿執筆時(95年1月)においても継続して行われている。

た、極めて研究的な色合いの強い活動でもあった。

ともあれ、その成果として、昨年度末に向けて、筆者は前述の仕様書の原案を作成した<sup>2)</sup>。この仕様書はテキストファイルで60KBに達するかなり大部のものになった。できればこれ以外に調査報告やこの仕様書の主眼点の解説等が文書化され(即ち本稿は1年前に作成されているべきだった)、それをもとに広く討論を行うという手順を踏むのがベターではあった<sup>3)</sup>が、時間的及び人的資源に関する強い制約があり、また、新旧委員会の引継時期だったこともあり、口頭による説明等によるかなり略式な経過で、ほぼそのまま見積書提出依頼の添付文書として採用され、結果的にはそのまま担当者との取引の基本書類としても使われ、現在に至っている。本稿は遅ればせながらそこで生じた情報不足を補うとともに、この新規システムについての学内外の理解を深め、また、ポストダウンサイジング時代の情報システムの在り方や難しさについて論じることにも目的とする。さらに本稿が今後、教育機関をはじめとする様々な組織の情報システム構築に関する方々の参考にもなれば幸いである。

なお、本年度(担当者決定後)は、インプリメンテーションの段階として、仕様書に基づく具

- 2) 主に筆者の力不足による作業の遅れや仕様策定過程での連絡不足等のため至らない点等が数多くあったことは関係諸氏にこの場を借りてお詫び申し上げる。
- 3) 但し情報システムの設計(デザイン)作業は、他の分野と同程度に「アート」的な一面を持つため、民主的であることが必ずしもよい結果を招くとは限らない。

体的な設計、調達、設置等の作業が、担当業者を主体として行われており、筆者はこれまでの経緯の延長で、引続き同委員会に所属し、担当業者に対する技術的窓口とも言うべき位置（即ち現在もこの更新に関して限りなく当事者に近い立場）にいる。その過程で浮かび上がって来た個々の要素技術やノウハウ、及びシステムの完成像やその評価等については、本年度の作業完了を待って、鋭意報告して行くことにし、本稿では概念設計のレベルでの話を中心とすることにしたい。

筆者はこの新規システムの開発コードとして、**NZIANCE** (Nagano University Advanced Network Computing Environment の略) と名付けている。これは少なくとも現時点ではあくまで非公式な名称であるが、本稿では便宜上この名称（とロゴ）を用いる。

## 2. 時代背景

### 2.1. 変化の速度

情報システムに関する技術の進歩の速さは（陳腐な表現であるが）「日進月歩」だと言われる<sup>4)</sup>。特に何等かの要素技術の出現が時代の要請とマッチして、社会全体にも影響を与えるような大きな変化を引き起こすことがある。古くはコンピュータの出現そのものがそうであったし、十数年前に現れた「パソコン」なるもの（以下では機器の名称としてPCと呼ぶ）もまた、計算機システムの形態をドラスティックに変えた。最近ではネットワークを始めとする、もともと計算機技術の外にあった技術を計算機に取り込むことにより（逆に計算機が通信システムに取り込まれていると考えてもいいのだが）、さらに大きな変貌をとげつつあり、その意味では現在が新たな過渡期であると言える。

当然、こういった変化は、大学など教育の場における情報システムの構築にも影響がある。また、関係者の考え方や制度がこの変化に追従できていないことも、この情報システムの世界では常であり、このギャップに由来する制約や困難も生

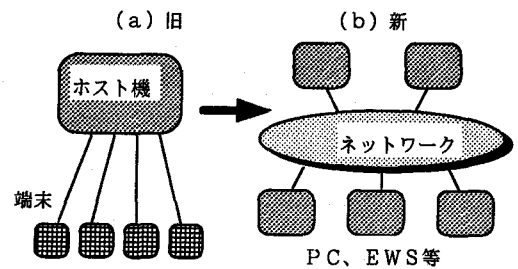


図1 形態の変化

じることがある。第3章以下ではそういった問題点の幾つかを取り上げ、**NZIANCE** の概念設計にあたっての筆者の考え方を述べて行くこととし、本章ではその前提となる現状認識を以下に要約しておく。

### 2.2. これまでの技術潮流

前に触れたように、PCの出現により<sup>5)</sup>情報システムの姿が大きく変化した。その変化は、以下の2つのキーワードで要約することができる。

#### 2.2.1. ダウンサイジング

IC技術等の進歩により、CPU、メモリ、OSといった、独立した計算機として必要な機能を一通り装備した機器が、ユーザの手元（デスクトップ等）に置くことができる程度に小型化し、また低価格になった。さらに製造、流通での位置付けが、プロユース向け或は産業用機器から、民生用機器へと変化し、価格、性能、信頼性の点での優位性を確保し、これがネットワークで接続されることにより、情報システムの基本的形態を図1(a)から(b)の形に変えた。コンピューティングは大型計算機室のアカウントを持つ一部の専門家から、一般ユーザの手に開放され、ユーザは、個人や小組織（部、課、室など）単位で自分のシステムを自律的に構築し、運営することができるようになった。

#### 2.2.2. オープンシステム

計算機の構成要素を互いに組み合わせて利用可

4) それでも足りなくて「秒進分歩」だと言う人もいる。その一方で本当に進歩しているのかどうか疑わしい面もあるが、少なくとも「変化」の速さは非常に速いといっている。

5) 実際には同時期にUNIXワークステーション(主に技術者向け；以下EWSと呼ぶ)が現れており、EWSとPCの両者が牽引してこの傾向を作ったと言えるのだが、ここではその歴史を詳細に見ることはしない。

能にするために必要な情報（インタフェース条件）が公開され、或は規格として文書化されることが、業界の慣習として定着した。これにより、ユーザが異なるメーカーの製品同士を組み合わせるシステムを構築することが可能になった。オープンでないシステムの場合、本体としてA社の製品Xを決定すると、ソフト、周辺機器、ネットワーク等すべてA社のX用しか使えず、即ちユーザの選択肢は極めて狭い（価格、性能、機能の点で不利であることが多い）訳だが、オープンシステムにおいては、ユーザは市場の中からかなり自由に構成要素の選択を行うことができる。

### 3. システムの寿命

#### 3.1. 陳腐化

前述の変化は、主として供給側のペースで行われてきている。一方の需要側は、必ずしもそのペースに律義に追従しなければいけない訳ではない。ある時点で入手可能な技術と製品をベースにユーザのニーズにあったシステムを構築し、継続して利用して行けばいい、という理屈は成り立つ。

しかし、現実にはそうはいかない。プログラムなしでは何の役にも立たない不完全な機械として生を受けた計算機は、その歴史を引きずってか、なにがしかの不完全さを常に抱えたまま、またそれを市場に容認されながら発展し、今日に至っている。つまり、どの時点をとっても、計算機システムはどこか不完全なシステムであり<sup>6)</sup>、常にその補修・改善を（利用の現場においても、供給側においても）行ない、完全なものに近づけていく必要がある。いや、それ以前にそもそも何をもって完全と見なすことができるか、その基準そのものが、技術や市場の変化に伴って変わって行くものである。特に大学の場合、社会の要請に答えられる情報技術の習得機会の提供という課題を抱えている訳で、従って、大学にとってのその基準（「望ましい情報システム像」とでも呼ぶべきか）は、常に社会の要請に対して相対的な位置関係をソフトしていく傾向がある。

その望ましい情報システム像と、使用中のシステムとのギャップは、放っておくと時間の経過と共に大きくなっていく。それがある程度を超えると、その時システムは「陳腐化」したと呼ばれる。この「陳腐化」は、機器の汚れ、傷み、老朽化といった物理的な老化とは別のものである。システムの物理的な寿命でなく、いわば論理的な寿命である。陳腐化の程度が激しくなると、

- ・システムが役に立たない
- ・ユーザが利用することを苦痛に感じる
- ・ユーザが使う喜びを感じられない（業務システムでなく教育システムの場合、これも無視できない要素である）
- ・上記の補修・改善のコストが極めて割高

といった問題が生じるようになる。陳腐化したシステムは生物（せいぶつ）<sup>7)</sup>に喩えると成長と新陳代謝が停止した状態だと言える。

システムの陳腐化は、概ね以下のような順序で発生する。

- (1) 心理的陳腐化：当該メーカーの製品の世代交代により、システム導入時は最新型だった機種が旧型、旧々型になり、現在売り出し中のシステムに比べて「見劣り」するようにユーザが感じ始める。
- (2) 性能的陳腐化：新型機種に比べてスピードや記憶容量等の点で不足し、ユーザが待ち時間や不便さを意識するようになる。
- (3) 費用的陳腐化：新型機種で提供可能な機能を、現システム上でユーザに提供できるよう、現システムを補修するとした場合に、非常にコストがかかるようになる（場合によっては新型機種に買替える方が安価である場合もある）。
- (4) 機能的陳腐化：メーカーによるサポートが停止されたり、現機種に対応可能なソフトウェアや周辺機器が提供されなくなり、補修・改善の道が閉ざされる（袋小路状態に陥る）。

この(3)、(4)の状態になると、機器の老朽化や、組織によっては減価償却を待たずして、機器の更新を必要とするようになる。ちなみに、本学の旧

6) 計算機を始め昨今の電子技術はユーザをベータテストに利用しながら発達してきているとも言われている。

7) ナマモノと読んでもかまわない。実際、筆者は情報システムをナマモノと考えて扱っている。

システム<sup>8)</sup>は、情報学科設立時より約6年強に渡って利用に供されていたが、やはり更新準備開始頃にはすでにそういう状況にあった。

### 3.2. 設計時の目標

機器の更新とは、別の言い方をすれば「買替え」であるが、通常、一応は動いているシステムを停止、撤去し、その後のスペースに新しいシステムを構築するという順序になり（部屋の移転を伴う幸運なケースを除く）さらに、システムの（使い方等の）変化にユーザが適応するための時間も必要とする。大学の教育システムの場合は利用度が大幅低下する期間（春季、夏期の両休暇）があるのが救いではあるが（勿論充分な時間とはとても言えないが）、学生や教員に与える混乱の大きさや、物品廃棄による環境への負荷という倫理的問題等を考えると、こういう作業はできるだけ避けるのが望ましい。そのためには、システム使用期間中は、補修・改善を継続的に実施していくことにより、一回の混乱を小さいものにするとともに、システムの論理的延命を図ることが必要である。

また、構築時には、陳腐化しないシステムを設計する必要がある。と口で言うのは簡単だが、これは即ち未来予知せよという意味であり、実際には不可能な話である。論理的寿命を5～7年と考えるとして、5～7年前に今の状況が予測可能だったかどうかを思い返せば納得できるだろう。（筆者は当時、5～10年のスパンで未来を考えることを生業とするような組織に所属していたのだが、昨今の状況は筆者にとっても「驚き」でしかない）。従って、極力「陳腐化しなさそうな」設

計をするよう努めることが設計者のできるせいぜいのことである（筆者の選択の結果は、その5～7年後に判明するのだろう）。

その際にポイントになるのは、恐らく互換性の問題であろう。例えばあるソフトウェアが機能的に不足或は問題があり、それが改善された新しい版が入手可能だとしても、それが現在あるハードウェア上で動作させることができなければ、そのソフトウェアの問題は解決されなくなる。つまり、今後長期に渡って利用され、また、市場において対応製品が供給され続ける見通しの高い、規格やプラットフォームを選択することが肝要だと言える。ただ、それは概して最新の技術、規格、或は製品であることが多いが、徒らに最新のものにこだわって選ぶことの危険性にも注意が必要である。それは費用の点で割高であったり、導入時のトラブルが発生しやすかったりする。場合によっては「枯れた」技術を採用するのが得策であることもあり、このバランスが設計者の悩み所であろう。

### 3.3. MISIANCE の場合

上記に関し、筆者が設計者として設定した指針を本節に示す。大学の広報記事等においては、

- ・マルチメディアの時代に対応
- ・最先端のテクノロジー

といった一般受けする（専門家には意味不明の感もある）華やかなキャッチフレーズが並べられる傾向があるが、デザイナーの立場から、是非鑑賞して頂きたいと感じるのは、本節のような微妙な部分についてである。

なお、大学システムの規模になると、構築そのものにかかり時間がかかる。その上、設計等の準備作業に多くの時間をさいた場合、システムが供用開始すぐに（或は前に）心理的陈腐化してしまう等の心配もある。従って、本節に挙げた以外に、設計者が情報収集や意志決定を迅速にできるような体制を作ること等、組織として配慮すべき事項もあると思われる。ただし筆者はその配慮の恩恵に預かる側の立場であったため、これについては詳しくは書けない。

#### 3.3.1. 標準品の採用

8) 実際にはこれがさらに進行した状況として、  
 (5) 維持運用のための年間コストだけで、同程度の能力の新しい機種購入費用を上回る額になる。  
 (6) 効能（金額換算）と処理費用（廃棄等）を比べると後者の方が高額になる（このとき、かつて「資産」だったものは「含み負債」に化けることになる）。  
 といった状況も発生している。しかし、だからといって旧機種購入時点において、それが大きな失敗だったという訳ではない。寧ろこれは情報システムの未来を予測することの難しさを示す1つのエピソードだと考えるべきだろう。

1社の独自規格のものを極力排除し、国際規格や業界標準として多数のメーカの支持を受けているものを採用する。そうでない場合でも、内部情報が公開され、或は互換品メーカが存在するなど、協調的競争状態にあるものを採用する。概してこういった規格や製品の寿命が長いこと、また、技術情報や補修部品の入手においても有利であることがその理由である。メーカの独自規格の製品（いわゆるオフコン周辺に多い）を購入して非互換性の畏にはまり、後の拡張に苦勞するユーザの例を、筆者は過去に（残念なことに現在でも）見てきており、この点については厳しく警戒している。

情報システムの世界はボーダレス化が進んでおり、従って日本だけでなく国際的に認知されているものを優先的に採用する。但し、漢字を扱う頻度が高いこと、また、マニュアルやメッセージが日本語で読めることが望ましい等の事情もあり、すべてこの基準通りにはいかないが、その場合は、単なる「日本語化」でなく「国際化」の一貫としての日本語化<sup>9)</sup>であることが望ましい。

### 3.3.2. バス的部分での贅沢

情報システムを図2のような2階層のバス構造と考へ、この2つの階層におけるバスに相当する部分には、現時点ではオーバースペックであるが贅沢なものを採用する。即ち、計算機のバスの規格としては32bitバスを持つことを条件とする。ただし、16bitのISAバスも今のシステムを構築するためには不可欠であり、従って両方のバスの併用が必要になる。実際にはEISAやVLバスは構造的にISAバスを兼ねているが、VLバスは実質的にビデオボード専用に近いため、具体的には、EISAまたはPCIが候補となるが、速度や将来性の点から、本命はPCIだと考えられる<sup>10)</sup>。また、ネットワークの伝送媒体として次の技術に対

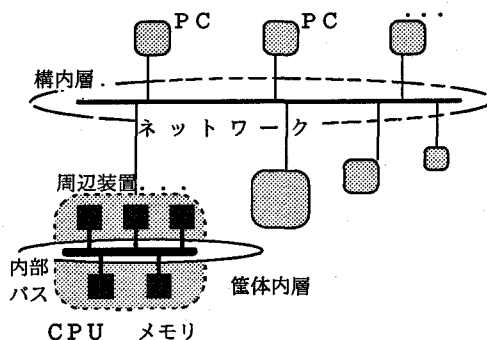


図2 2階層のバス構造

応可能なものを用いる。建屋内はカテゴリ5のUTP、建屋間は芯数に余裕を持たせた光ケーブルを敷設する。

この2つのバスの部分は、陳腐化した時の影響が大きく、従って、論理的に長寿であることが望まれる部分であることがその理由である。筐体内バスを新しい規格に変更するということは、それに直接接続する部品がすべて役に立たなくなる訳で、それは即ち計算機機本体をまるごと買替えることに等しくなる、影響の大きい部分であり、また、ケーブルリングの更新の場合は、大掛かりな撤去及び敷設工事を必要とするからである。

### 3.3.3. 周辺部分での日和見

前項とは逆に、寿命に関して要求の低い部分については3.3.1項の規格条件は考慮するものの、新しいものにはこだわらないこととする。仕様書作成時点ですでに次の世代のものが出荷或は発表されていたものとして、OS(Windows NT等)、ネットワークの方式(FastEthernetやATM)、MPU(Pentium等)といったものがあるが、仕様書にはそれを条件として書かず、選択を業者に任せる(業者は経済的条件および自社での技術蓄積状況と今後の展開方針等を総合判断して決定することになるだろう)。概ね、以下の何れかに該当するものに関しては、日和見することとする。

- a) モジュール化された構成要素の1つで、前述のバスの位置になく、従って、陳腐化したとしても、その該当部分だけの更新が可能で、他への影響が少ないもの。
- b) サーバ機やネットワーク機器等、全体での台数が少ないため、少ないコストで更新が可能な

9) ワープロ等の自然言語関連ソフトにまでこの基準を適用してしまっていることについては議論の余地があるだろうが、5.3節脚注の事項も考慮すべきだろう。

10) 仕様書リリース時点でクライアント機向けの適切な価格のPCI機が国内でまだどこからもアナウンスされていなかったため、この条件を盛り込むことは実は大きな冒険であった。

もの。

- c) 現在ホットな分野で、製品のライフサイクルが短く、この先短い期間で大きく変わることが予想され、どのみち3~4年以上先の見通しが立ちそうにない(従って、これは比較的早期に陳腐化することを覚悟するしかない)もの。

### 3.3.4. ディスプレイの贅沢<sup>11)</sup>

ディスプレイに関しては、一般用途(即ちCAD等の特殊用途でない)では現時点では最高レベルのもの(17inch、水平解像度(NI)1280dot)を条件とする。これは、前項の3条件について、以下のようにいずれも該当しないからである。

- a) モジュール化されておらず、交換するとしてからディスプレイまるごとになる。  
b) ユーザ機の台数分必要。  
c) 下記のことから、今後の変化は緩やかになると予想される。

1. CRT そのものが技術的に成熟してきており、性能的にも限界に近い。
2. 今後、解像度が上がる場合(縦横とも上がるとして)、伝送周波数等に関する性能要求は解像度上昇の2乗に比例して上昇することになり、技術的には困難が大きい。
3. 17inchでは、現在の解像度を超えても、人間の視力の方がそれを活かしかねないし、日本のオフィス事情や住宅事情を考えるとそれ以上のサイズのもので普及することも考えられない。

現時点ではCRT以外の技術の採用は実質的に考えられないが、他の分野での表示技術の開発は現在ホットに行われている。従って、今回導入するCRTが将来撤去される時は、次にこの場所に置かれるものは、CRT以外の何か(液晶等のフラット型か、でなければ立体映像か)であろう。

## 4. 調達方法の問題

### 4.1. よくある落とし穴

第2章に述べたような変化により、情報システムに関してユーザの自由度と権限は飛躍的に増大

11) この部分は、業者決定後の仕様変更により要求レベルを若干上げた部分であるが、画面寸法や解像度については仕様書段階で決定していた。

した。知識を持つユーザは、自分で情報システムを構築、運営することができる、そういう時代になった。これは基本的には、望ましいこと、である。しかし、多くの分野において「技術」が陥りやすい落とし穴がある。実際には技術の側だけに責任がある訳ではなく、その技術を取り込んだ社会の方からの圧力が直接の原因になるのだろうが、ともかく、「〇〇することが可能」になった後、短い期間のうちに「〇〇するしかない」という状況が作り出されてしまうことが往々にしてある。つまり「〇〇する」以外の選択肢が消失してしまうのである。

例えば、我々の身近で(特に地方では)、家用自動車の普及により公共交通網が破壊され、自動車以外の移動手段が失われつつある(この場合、〇〇には「移動手段の自己調達」が入る)など、こういった現象は我々の社会では枚挙に暇がない<sup>12)</sup>。計算機の場合でも、自分で構築や管理が可能を通り超して、自分でやるしかないという状況が発生しがちである。

### 4.2. システム管理者の状況

大型機の時代には、計算機は高価なものであり、それを購入(或はリース等)してしまったり、それを最大限に活用して元を取る必要があり、そのための(一般にセンターと呼ばれる)担当部門が用意された。現在、それらは必ずしも必要ではない。そのため、担当部門は用意されないことが多い。「『必要』でなくなったことは行われぬ」という法則があるのかも知れない。或は、パソコンを買って並べればシステムができる、と思込んでいる組織管理者が多いのかも知れない。現在、多くの組織において、計算機やネットワークの世話をしているのは、専門職でなく、(大学では研究者や事務職員など)自分の本来の仕事を持つ少数の個人によるボランティアである<sup>13)</sup>。作業は業務時間内に行うとしても、それを支える調査

12) 情報技術もこういう危険性を多分にはらんでおり、関係者の一人として、この種の現象に目を離す訳にはいかないし、いずれこれについてもじっくり調査してみたいと考えている。

13) 経験則によれば、この人選は、関数  $f(\text{人}) = (\text{関連知識} \times \text{好奇心}) / (\text{役職位} \times \text{押しの強さ})$  で決まるようである。

活動（一種のシャドワークだろうか）は個人の時間に行われ、ペイされない場合が多い（大学によっては院生の活力なども利用されているが、それとて概して彼らの研究テーマ外の雑用である）。

勿論、この活動はその個人にとって必ずしも不利益なことではなく、活動を通じて得られる知識というご利益もある。つまり勉強になるのである（筆者にとってもそうであった）。ただ、この勉強には、学校制度の持つ最も巧みなしかけである「卒業」が用意されていない、これが問題であろう。彼らは（筆者も）運よく後任が見つかるまでこの業務から解放されることがないのが常である。計算機の世話をする仕事（計算機ユーザの世話をする仕事でもある）は、決してなくなるらない。寧ろ、多くの場合増大する傾向がある。パソコンの時代になって十年余り、ボランティアがボランティアでいることに疲れ始めている<sup>14)</sup>。

#### 4.3. サービスのアンバンドリング

十分なボランティア要員を確保できない組織は、供給側（メーカーや販売店）のサポートをあてにする。機械を買うと、オマケとしてついてくる技術者が面倒をみてくれる、という考え方がかつては成り立っていて、供給側もその要求に応える体制を作ることができていた。ハードウェアが高価であるため、サポート等のサービスにかかる経費はその中に吸収してしまうことができたからである。

ダウンサイジング時代には、そうはいかない。ハードウェアの低価格化と人件費の（相対的）高額化により、コストの大小関係が、

ハード > ソフト > サービス  
から

ハード < ソフト < サービス  
に逆転してしまっているからである。約20年前から、狭義のプログラム製品としてのソフトウェア（ここでは狭義のソフトウェアに限ってソフトと略記する）のアンバンドリング（ハードウェアのオマケでなくなる）が始まり、すでに定着しているが、さらに今、運用、サポートといった広

義の（プログラムに限定しない）ソフトウェアも別建ての価格を持つようとしている。

しかし、どうやら現在、購入者側が明確にそれを意識するにはまだ至っていないというのが実状のようである。機器調達にあたっては、今でもその仕様書に、形のあるもの（ハードやソフト製品）の名称が並ぶことが多い。これは、供給側から提供される情報（カタログ等）の方にまだ不十分な点があるのも一因だろうが。いずれにせよ、システムを構築（次節に述べる）したり管理したりするのに必要な（手間等の）コストは、末端供給者の出血サービスと、ユーザ側ボランティアによって吸収されることにより、かろうじてシステムが成り立っている状態とっていいだろう。これもそろそろ限界に近い。我々は次の方法を探らなければならない。

なお、本章ではダウンサイジング+オープンシステムという方向への動きの結果として生じた問題点を幾つか指摘しているが、この方向が間違いだったと言いたい訳ではない。計算機及びそれに関する情報を自由に入手し使用できるようになった。これは大きな進歩であり、かつてのように少数の大企業の内部に情報が集中し隠蔽された状態には、決して戻るべきではない。本稿は、現在の状態を乗り越えてさらに前に進むことの必要性を主題にしているのである。

#### 4.4. インテグレーション

オープンシステムとは、マルチベンダーによるシステムのことでもある。ハードメーカーとソフトベンダの住み分けや、専門化が進んでおり<sup>15)</sup>、一社の製品でシステムを構築することは不可能になってきた（前述の法則がここでもあてはまる）。複数のメーカーの製品を組み合わせるシステムを構築する必要があり、ここに以下のような「システムインテグレーション」に属する仕事が発生する訳である。

- ・市場に提供される豊富な製品群から目的システムに最適のものを選び出す（選択肢が幅広いという利点は、残念ながらこの作業が膨大

14) これは組織管理者が考慮すべき問題で、一介の設計担当者がなかなか口出しできない領域なのだ。

15) OEM 等により幅広い製品ラインナップを持つメーカーもあるが、そうなるともはや商社と呼んでも誤りではない。

であるという欠点にもつながる)。

- それら相互間の適合性の確認。
- 納期の調整や、ケーブル類の手配等の準備作業。
- トラブル発生時に障害の切分けと該当メーカーへの連絡。

これは作業量を増大させるだけでなく、計算機、ソフト、デバイス、ネットワーク、及び(教室システムの場合)映像技術といった、複数の領域にまたがる広くかつ深い知識を必要とする。また、同じ規格やインタフェースに基づく製品相互間においても、相性のよいものが存在し、性能低下や不適合が生じるケースもあり、こういったリスクも誰かが背負う必要がある。

ユーザ組織の中で生半可な知識を持つ者が SE の真似事をして、製品名や部品名を並べた仕様書を書いて物品購入をすることが、盛んに行われているが(筆者もよくやった)、その結果、相互接続ができないだのアダプタの型番が違うだのという不適合を発生させてしまうことが(マルチベンダーでなくても)往々にしてある。この場合、ユーザはどこにも責任をとらせることができない。販売業者に「泣きつく」か「泣かせる」か自らが「泣き寝入り」するか、いずれにしてもどこかに損失や摩擦が発生する。このリスクは、ユーザ側で背負うべきではないだろう。

こういった作業とリスク、及び前項のサービスといった、広義のソフトウェア全般を請け負うビジネスとして、いわゆる SI (システムインテグレータ)<sup>16)</sup>がある。我々ユーザー組織の者が、これまで本章で述べてきた問題を解決或は回避するためには、SI による(商品としての)サービスを活用すること(及び、活用のしかたを覚えていくこと)が最も適切だろう。

勿論、ユーザ側が勝手に選定し、或は調達した個別の機器や部品に関して、無制限に SI がサポートしてくれる訳ではない。サービスのみのビジネスはなかなか成立させにくい<sup>17)</sup>し、それぞれの SI 毎に、取り扱い技術の範囲やメーカーとのパイ

プの太さ等の、得手不得手がある。従って、無用なリスクや摩擦、或は人材の疲弊を減らし、健全なシステム構築を進めるためには、商品の調達やさらには可能な限り選択も含めて、できるだけ一括で発注することが、人的資源の乏しい組織にとっての(少なくとも現場からの視点では)最良の策ということになるだろう。こうしたことを考慮して、*MZIANCE* では、個別の製品の調達やメーカーとの直接取り引きを避け、商談の対象を SI 業者に限定し、一括発注の方向で仕様書を作成した<sup>18)</sup>。

これに加えて、本学のような地方大学では、情報関連技術の地域格差にも気を配る必要があるだろう。ハードウェア製造やパッケージソフト開発においては、例えば信州地区にも優れた技術蓄積がある<sup>19)</sup>が、インテグレーションに関する情報の豊かさや技術者の層の厚さにおいては、残念ながら東京等の大都市圏と地方との間に大きな格差がある。だからといって首都圏の SI 業者にすべてを委することは(当面のリスク回避にはなるだろうが)、必ずしも得策ではない。次年度以降のサポートにかかるコスト増(技術者の交通費や交通時間も大抵こちらの負担になる)のこともあるし、また、地域経済にとっての損失でもある。大学の人間としては、寧ろ、地域に SI 業者を育てていくという気概を持つことも必要だろう。

#### 4.5. 機能を購入する

##### 4.5.1. 発想の転換

我々が何のためにシステムを導入するのかを改めて確認しておこう。それは、実習室に機械を並べたいためではない(大学の客寄せ看板としてその意味はゼロではないが)。実習をするためである。実習のためのソフトウェアが健全に動作すること、が本筋である。我々が欲しいのはモノでな

16) 通産省による認定制度があるが、ここでの SI は、認定 SI に限ってはならず、インテグレーションに類するサービスをビジネスとする業者全般を指す。

17) 実際、SI 業者もまだハードウェア中心の見積書しか書けないでいることが多いのだが、これは発注者受注者双方のこれからの課題ということだろう。

18) 現実には映像提示系等に関する技術的及び価格的な見通しを付ける作業が遅れたため、一部のサブシステムについては別途発注となった。

19) 実際、県内で開発或は製造された製品が、結果的に *MZIANCE* に数多く使われている。



く機能なのである。或は、コンピュータではなく、コンピューティングである。結果的にはモノが設置されてしまうにせよ、我々の要求は、極論すれば、機能さえ達成されるのであれば、筐体の中に CPU やディスクでなく妖精さんが入っているというものだって構わない訳である。

にも拘らず我々は、システム導入に関する取引に関して、「買い物」とか「物品購入」といった言葉を使う。我々の求めるものはモノではなくコトなのだが、こういった言葉使いは我々の発想転換を妨げる傾向がある<sup>20)</sup>。これからは例えば「買い物」という言葉を使っていくのが適切かも知れない。

#### 4.5.2. 実現における選択肢

プログラミングの世界では、“There’s more than one way to do it.”<sup>[2]</sup> というような言葉<sup>21)</sup> がよく使われる。1つの目的を実現するために方法は幾通りもある、という意味である<sup>22)</sup>。これは情報システムの構築においても広くあてはまる、のみならず、ある機能を実現（インプリメント）するために、ハードを使うか、ソフトを使うか、或は場合によってはそういった仕掛けを使わずに人力で行うか、といった、異なる形態に渡った選択肢が存在する。特に最近ではエミュレーション技術が発達し、ある計算機がソフトウェアにより別の種類の機械の「ふり」をしたり（端末や OS のエミュレーション）、ある装置や回路（FPU、ディスク等）を内蔵しているように「見せかけ」たりすることができる。このうちどれを選び組み合わせるかは、性能やスペース、コストといった条件を勘案して総合的に判断することになる。こうなるともうディレクタントの手には負えない。

#### 4.5.3. 仕様書の記述

こうしたことから、*NUANCE* では、機能に関する要件を仕様書に網羅することに重点を

置き、これを個々の技術や製品といったインプリメンテーションのレベルにブレイクダウン（具体化）した記述は極力避けた。つまり、製品名を羅列したものでない仕様書を目指した。実際、特定のメーカー名や製品名に関する言及は少なく、その代りに規格の名称や機能的条件に関する記述が多い（その結果として長い文書になった訳だが）のがこの仕様書の特徴だろう。ただし、機能を羅列するよりも特定の製品名を挙げて、「〇〇のような機能」「△△相当品」と書く方が（書く側にも読む側にも）手っ取り早い場合は多く、そういう記述を行なった箇所もある。実際、我々は情報システムのイメージを（needs でなく）seeds の側から（つまり実際に市場にある商品をもとに）形成して行く傾向はかなりあり、従って、筆者の作業も、要求から製品へのブレイクダウンを省略したというよりは、製品群からのイメージを要求事項の形に再構成する、抽象化の方向での作業を余分に行なった、というのに近い。

#### 4.5.4. 性能条件

この作業により機能要求は記述することができたが、一方の性能要求に関しては、その条件を具体的に記述することは非常に困難であった。性能に関する我々の要求は、要約すれば「あらゆる場面で（待ち時間等の）支障なく使えること」であるが、どの程度を以って支障と見なすかは主観的な領域である。（後述のように）取引の条件に使う場合には、客観的な数字を必要とする。また、その「あらゆる場合」について性能を検証することは、納入側にとっても発注側にとっても不可能である。

かつて、計算機の性能がほぼ CPU の能力で決定していた頃があった。計算機を文字通り「計算機」として（即ち CPU が主役となる用途で）使っていたからであり、また、一社で計算機全体を設計・製造するため、CPU 性能を殺さないトータルな設計が可能だったからである。そのため、CPU 性能を数値化する指標（ベンチマーク）は古くから用意されており、これを条件として記述すれば事足りた。現在、計算機の多くの用途において、CPU は性能決定要因ではない。メモリ、バス、ディスク等の様々な箇所での性能的なネックが

20) 多くの「図書館」が情報電子化の時代になってもペーパーメディア専業からなかなか脱却できないのも、多分にその名称に自己規定されてしまっているからであろう。

21) 恐らくこの分野に限らず適用される真理だろうが。

22) 例えば教育の現場ではこの事実を忘れがちである。自戒しておきたい。

生じ得る。そのため、これらを含めた計算機の総合性能を数値化する指標も最近は使われつつある。

ただ、それらはまだ単体計算機を対象としたものである。ネットワークがボトルネックになるケースは、まださほど考慮されておらず、ネットワークシステム全体の性能指標は、まだまだ今後の課題という状態である。単体計算機の場合（シングルプロセッサに限れば）、その構成は「バス構造のバリエーション」の一言で括ってしまえる、ある意味で単純なものだが、一方、ネットワークの構成（コンフィギュレーション）は、多種多様で、つまり様々な接続の形が存在し、これが数値化を困難にする一因だろう。その上、多種多様な構成が可能であるゆえに、個々の構成機器の性能を活かすも殺すも構成次第となり、つまり、構成の決め方の巧拙がシステム全体の性能に与える影響が大きい。その意味で、ネットワークの構成が、インテグレーションの作業の中で、現在最も難しい問題であろう。

本章で述べて来たように、筆者はこの問題をユーザ側で解かないストラテジをとった。ネットワークの方式、或はサーバの機種や台数といった具体的な事項は一切記述していない。要は、性能的に不足がなければ何でもいい、インプリメンテーションは SI に任せましょうという姿勢である。しかし、次章に述べるように、**NIANCE**はそのネットワークを中心に据えたシステムであり、実際、ネットワークが全体の性能を決める可能性が高い。そのために有効な指標が用意されていない、これが、非常に困った所であった。また、性能条件を数値として記述するための現実的な値を見積るための、目安となる経験値や参考資料等も、乏しいのが現状である。

筆者は結局、ユーザ機の起動、ソフトウェアの起動といった、待ち時間の発生しがちな典型的なケースを想定して、そこでの待ち時間の最大値を条件にする、という記述を行なった。数値としては、パソコンをスタンドアロンで動作させた場合

の経験値をもとに、それよりも心持ち甘い数値<sup>23)</sup>を設定した。かなり略式の指標である。このあたりの有効な方法論を探すことは、デマンドサイドの情報工学の今後の重要な課題であろう。

#### 4.6. 設計の過程

##### 4.6.1. 既製品利用の原則

こうして作成された仕様書は、機能的および性能的な要求事項を取りまとめた形のものになっている。しかし、ユーザ側で必要だと思われることをただ書き並べるだけで仕様書が出来上がる訳ではない。問題は実現可能性である。仕様書に「ないものねだり」を書く訳にはいかない。情報システムの場合、速度等の性能に関して非現実的な要求でなければ、必要に応じてカスタムメイドのソフトやハードを利用することにより、大抵の要求は実現可能であり、「ないもの」であるケースは希である。しかし、コストという要件が加わると、話は別である。仕様書とは、つまり「ここに書かれた要求を満たすシステム構築を適当な価格で請負ってくれる業者」を募る文書である。適当な価格での提案が出てこなければ、話が始まらない訳である。

情報システムの世界では、開発コストが製造コストに比べ格段に大きく（特にソフトの場合は後者はほぼゼロである）従って、既成品とカスタムメイドでは、そのコストは（洋服とは比較にならない）桁違いの差になる。**NIANCE**では約100台のユーザ機を導入したが、この数字はユーザにとっては大きな数字でも、開発側にとっては「小口（こぐち）」である。他に大きなマーケットが見込まれるケースででもなければ（それなら我々に要求される前に開発を始めてるだろうし）、開発側がそのコストを負担するとは考えられないから、従って、このコストは、開発時のリスクと共に発注側で負担することになる。適切な価格を望むなら、余程のことがない限り、カスタムメイドを使うことは避けるべきである。

23) 実際にはネットワーク経由のデータ転送の方が高性能であることは珍しい話ではなく、また理屈から考えても当然なのだが、ここは筆者にも経験の乏しい領域であり、強気に出られなかった部分である（商談に関った業者の SE 諸氏はそうは思っていないだろう）。

さらにカスタムメイドの場合、後々の保守についてもユーザ側で面倒を見なければならない。勿論、ソースコード（ソフトの場合）等の技術情報を自ら保持することにより、供給者の営業方針に影響されずにそのソフトを改良していける自由度は、ユーザにとって魅力ではある。しかし、それだけの資金やマンパワーをユーザ側で維持できるケースは希であり、特にソフトに関しては、市場でもまれている流通ソフトや、フリーソフトの類の方が、着実に（かつ素早く）進歩している。その分、カスタムメイドのソフトは、陳腐化するのも早く、ゴミを作り出してしまう可能性も高い。自ら育てていく覚悟と裏付けがユーザ組織の側になければ、割に合わない活動である<sup>24)</sup>。結局、今の時代の情報システム構築の技術とは、市販のものを組み合わせ、使いこなす技術だということになる<sup>25)</sup>。

ただし、これはユーザ側にとってはあまりに悲しい結論である。ユーザの要求と市場に提供される製品群との間には常にギャップが存在する訳だが、そのギャップに対して、常にユーザ側で折れる（要求を撤回する）ことが求められていることになる。現在、価格面においてはユーザが強い立場になって来たものの、どんな製品が市場に出るかに関しては、まだまだ供給側主導である。組織を超えたユーザ同士のネットワークが未発達<sup>26)</sup>で、消費者連盟的な共闘が行われていないことも、その一因であろう。これもデマンドサイドの情報工学の今後の課題かも知れない。

NUANCE においても、適当な製品が現在の市場に存在しないために仕様書への掲載を断念せざるを得なかった要求事項が多々ある。その一方で（設計者の意地のようなもので）該当製品や

前例がないことは承知の上で要求を貫徹するという冒険を行った項目もある。これは、ユーザ主導の情報システム構築に向けての1つの挑戦という意味を持ったものであった。その主要項目（その一部については、担当 SI が苦しみながら作業しており現在未完の項目もある）については、次章に報告する。

#### 4.6.2. 情報源

前項の原則により、設計過程は、市場にある技術と製品の調査から始まる。調査すべき内容としては、技術および製品の動向、各要素技術及び製品の機能や価格、問題点、ユーザ事例、今後の方向などがあげられる。さらに、それらを組み合わせた場合に発生する可能性のある問題等についても、設計に先立って知って置く必要がある。この調査は、まともに取り組むと膨大な量になる。調査手段としては、実機を操作し検証する、展示会に出向く、雑誌や Internet からの情報収集、電話による問い合わせ等の方法が考えられ、実際に筆者もこれらの方法を併用した。

しかし、これでは不足である。手間の問題と共に、メーカーとの直接のパイプを持たないことも、ユーザ組織の人間にとっては、足枷になる。実際、製品の価格に関する情報は（民生ルートが確立した一部商品を除いて）、ユーザに直接届くことは少ないし、（特にソフト等について）流通の都合や方針によって、かなり変動幅が激しく、ユーザの手におえない領域である。したがって、SI 等の協力が、調査段階から必要になる。すると、その協力をどの SI に依頼するか、という問題になる。

#### 4.6.3. 決定のプロセス

ここに、依存関係のループ（計算機科学<sup>27)</sup>では「ブートストラップ問題」として、或は我々の身近では「鶏と卵」や「3すくみ」として知られる問題）が、以下のように2重3重になって発生する（図3）。

a) 予算を決定するためには業者からの見積りが必

27) 蓄積プログラム型計算機の黎明期から存在する問題なので、ここでは「計算機科学」で正しい（5.3節参照）。

24) プログラムが余っている、域は調査能力がない、といった事情を隠して徒らにユーザにカスタムメイドを薦める SI には要注意である。  
 25) 尤も、この結論は、トップダウン（要求の側から）とボトムアップ（利用可能資源の側から）との両方向からの事情の摺合わせが必要だ、という、（特に分野を限定しない）設計者の普通の発想法を述べたにすぎないとも言えるのだが。  
 26) 計算機の分野で現在あるユーザ相互間組織は、殆どが特定のメーカーや製品のユーザ会であり、メーカー主導のもとに情報交換を行っている場合が多く、市場全体への働きかけはないといっている。

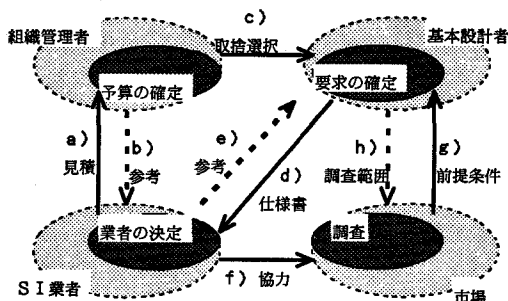


図3 依存関係のループ

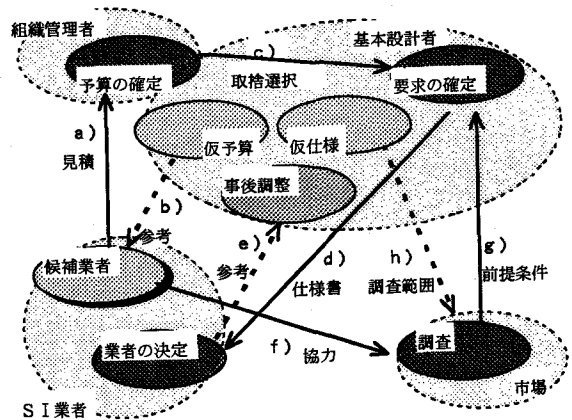


図4 ループの解消

要。

- b) 業者は予算に応じた提案を出すことを基本的な仕事のパターンとしている場合が多い。
- c) 要求内容の取捨選択は、予算に合わせて行う必要がある。
- d) 業者を決定するためには、仕様書とそれに対する価格提示という、公正な手順が必要（私立大学では入札を義務付けられてはいないが、それに準じた決定プロセスが主観やコネの入らない最も健全な方法だろう）。
- e) 担当 SI 業者の方針や技量等によって、要求事項の実現可能性にも変動が生じ得る。
- f) 業者の（調査等の）サービスにも対価が必要であり、その依頼先は調査に先立って決定している必要がある。
- g) 要求の確定のためには十分な調査が行われている必要がある。
- h) やみくもに調査するのでなく、要求事項に基づいて調査範囲を絞らないと効率が悪い。

このうち、いったいどれから順に決定していけばいいのか。勿論、これはどの組織でも発生する問題である。

**INIZIANCE** 準備段階においては、図4の手順をとった。まず、最初に仮の要求事項と仮の予算を提示した（これで図3のリンク a と h がループから外れる）。また、準備段階（担当 SI 決定前）において、「商談に参加できることを条件に」という、いわば交換条件を提示することにより（筆者はその権限を持たないのだが、「各業者の技量を評価する」という名目をつけると話が通ってしまう）、複数の業者に調査への無償の協力と仮見

積の提出を依頼した（これでリンク f も外れた）。これにより、話しの順序関係を整理することができた。3者（市場を除く）の間をオフィシャル（または半オフィシャル）な情報が2周回ることになる。

ただし、その結果、調査協力と営業活動を兼ねた複数の業者が、個別に入れ代わり立ち代わり筆者の所に入出入りすることになり、別の苦勞を背負込むことにはなった。ここで協力依頼する（または参加を申出る）業者の数がやたら多いと、その苦勞は徒らに増えるし、また、少ない場合には、情報不足に陥ることもさることながら、本番の商談において（応札するかどうかは各業者の選択であるから）、適切な提案が一件も提出されない不安が発生する。かといって、業者との対応を公開説明会的に一括で行った場合には、重要な情報がこちら側に流れてこない可能性が大きい。こういった雑事と、自分の業務をいかに両立させるかが、ボランティア管理者の悩みの種である。と同時に、このユーザ側の過程については整理されてもおらず、ノウハウの蓄積もない、この現状もまた問題であろう。

なお、本章ではユーザ組織にとっての課題を（偉そうに）書いて来たが、筆者の作業がその目標を十分に達成していると言明するものではない。本稿は筆者が目標とした方向について述べている（それに対して不十分だった面は自ら多分感じている）と共に、問題の指摘とユーザの結束の呼び掛けも行なっている訳である。

## 5. システムの基本的形態

### 5.1. 不確定性

教育機関においては、前述の「卒業」という巧みなしかけにより、利用者への入替えが（部分的にしても）定期的に行われるのが特徴である。また、一般的に「自分の席がない」のも特徴であろう。これは計算機利用に関する話であるから、大学に限った状況ではない。尤も、小、中、高校で各教室にワークステーション（パソコンや端末などユーザが直接操作するもの、以下 WS と書く）が配備される1人1台の時代にでもなれば話は別であるが、つまり、個々の WS は、（時分割で）共有されるというのが教育機関のシステムの最大の特徴である。勿論、会社などのオフィスにおいても、1人1台の WS 配備でなく、共有型のシステム作りをしている所は多いだろうが、だとしてもその共有の範囲はせいぜい部や課といった狭い範囲に限られており、それに比べると、教育機関では、他に誰が使うのかわからない、即ち、より匿名性の高い共有が行われる。ここに、2つの不確定性が発生する。

- (1) (WS の側から見て)誰が使うかが不確定
- (2) (ユーザ側から見て)どの WS を使うかが不確定

### 5.2. プログラムからの訣別

ここでは2つの意味で「プログラム」という言葉を用いた。いずれも大学の情報システムを考えるにあたって、ともすればこれらに縛られてしまいがちになる古い常識である。と同時に、ここから2つの（一見）相反する要求事項が導かれてもいる。

#### 5.2.1. プログラミング

計算機を使うことは、即ちプログラムを作ることであるという時代があった。計算機に対してユーザが許される唯一の行為が、プログラム（とデータと JCL）を「インプット」すること、という時代があり、また、将来プログラマーが xx 万人不足すると騒がれ、教育機関にプログラマー養成の期待がかけられた時代もあった。筆者自身も計算機とのつきあいの入口はプログラミングだっ

た。

しかし現在、様々な既存ソフトが利用可能となり、自分で（或は自社で）プログラムを作らなくとも計算機を役立たせることは可能であり、それらを使いこなすための技術や知識（前にも触れた広義のソフトウェアである）の重要性は高まっている。これは、ハードやソフトの供給者側およびユーザ側の双方について、人材養成の重要なテーマであるし、教養としても然りである。計算機は、プログラミングの道具であり対象であった訳だが、それ以外の用途がこれまでに数多く開拓され、また今後も限りなく広がりがつつある。

だからといって筆者は、知的活動としてのプログラミングの意義と必要性を否定するものではない。ただ、我々教壇に立つ人間が、ともすれば自分が学んだ時代の感覚そのまま、或は計算機の用途の多様化の流れに追従し損ねて、つついプログラミング偏重の教程作りをしてしまいがちである、その危険性<sup>28)</sup>については心に留めておきたい。なお、この多様化した用途の中で、最も注目すべきは、通信網を通じて提供される様々なサービスに対する窓口、としての計算機の役割であろう。計算機は今や1つのメディア<sup>29)</sup>であり、或は情報通信網の1部品であると言ってもいい。とすると *NUANCE* の主役はネットワークである。その内部でその様々なサービスが運用可能であること、外部のサービスが利用でき（大学であるからには Internet アクセスは必須である）、ま

28) 第3章の表現を用いれば、(教員やシステム管理担当者における)「人の陳腐化」であろう。これを防ぐためにも、システムの継続的構築は重要である。

29) メディアとは情報の発信受信両方の道具を意味する言葉であるが、その発信側をビジネスとして目論む業界の誘導なのか、世間では、メディアとしての計算機としては、その受信側の機能だけ着目されてしまう傾向がある。そのため、情報発信のための機器類（主に入力機器等）の技術と市場はまだ成熟しておらず、その結果、スタート時の *NUANCE* も、予算との兼ね合いもあり、残念ながら機能としては受信側に偏ったものにならざるを得なかった。例えば音声データに関しては、音声入出力ボードとスピーカーまでは設置できたものの、マイク（或はヘッドセット）の設置や電話システムとの融合（いわゆるテレフォニー）はできなかった。画像や映像についても同じ様な状況である。

た、今後学内で開始されることが予想（期待）されるサービス（図書館や事務局）も利用できる必要がある。

さて、こういう背景から、システムは幅広い用途を提供するものであることが要求されている。これは、ソフトの豊かさで決まる、とっている。市場に様々なソフトが提供されていること、また、それが安価に入手できること、これが重要な要件である。それだけではない。情報技術全体がまだ進化の途上であり、また、そのために我々のシステムも継続的構築により進化していく必要があることは、すでに第3章に述べたが、この「進化」が現実にも可能であること、これが重要な点である。

新しくソフトを入手したとする。これを100台の計算機にインストールしたい。フロッピーやCDをがちゃがちゃ差し替えて、パラメータを入力するという、同じ操作を100回繰り返す、これを誰がやるのか。何かバグでも出れば、さらにもう100回。これを何か小さな変更の度に繰り返す、これはどう考えても現実的ではない。何等かの集中管理のメカニズムを使って、この構築及び管理の手間を最小限に押さえておくこと、これは必須の条件である。

### 5.2.2. プログラム学習

本来、学習とは能動的かつ自律的な行動である。関心を持ち、その関心の対象に向かって行動し、その結果として知識なり智慧を得る、これが自然な学習の姿である。その関心の持ち方、行動の道筋等には、人により十人十色とも言える個人差があり、それに応じた学習機会が提供できればそれが理想であろう。しかしながらこの理想の実現のためには残念ながら資源（人）が絶対的に不足している。そこで教育の現場では、仕方なく教室全員で一斉に取り組む教材を天下りの設定し、せめてその教材ができるだけ多くの教室員（学生）に適合するよう、工夫と苦勞を重ねる訳である。

実習の場では、この教材が、ともすれば学生の行動を細かく規定してしまいがちになる。まず○を××して、次に△△を□□して…。こうなると、これそのものがプログラムである。このプロ

グラムを実行するのは計算機ではなく学生であり、これは、学生を計算機（程度の知性）扱いしていることになりかねない。教員の立場からすると、これは、最初の高い敷居（または塀）を乗り越えて、自律的に行動できるレベルの場所まで彼らを導くための、必要悪のようなものなのだが、それとは別に、教える側の都合で、不必要に彼らの行動範囲を限定してしまう傾向もある。その根底に「変なことをされたら（私が）困る」という、おそれが存在している場合がある。計算機の持つ（潜在的）状態空間は非常に広く、一個人でとても把握できるものではない。実習者が予想外の操作をして、システムがダウンしたり、授業や他の人の作業に支障が出たり、或は、復旧の方法がわからなくて、教員側が冷や汗をかいたり、そういう事態を恐れる気持ちは、感じてしまうものである<sup>30)</sup>。

勿論このプログラム型学習過程も時として必要であったり効果的であったりする。しかし、それとは別に、計算機の前に（そもそもここに足を運ぶことから個人の自由意志であることが望ましいが）座った各学生が、計算機内部（？）に形成された情報空間と状態空間の中を、自由に「遊ぶ」<sup>31)</sup>ことができる、そういう場もまた必要であろう。いわば「公園型」学習機会の提供である。そこでは、システム管理側や教員の都合による制約や禁止事項（「このファイルは消してはいけない」「このディレクトリの下は見えてはいけない」「このプログラムは起動してはいけない」など）は存在しないことが望ましい。そこで行われる活動は、壁紙の変更を入り口とした「自分の環境」作り等の（概してマニアックかつ非生産的な）行為や、誤ってシステムファイルを消してしまつてうろたえること、も含まれる。効率が悪いが、これも学習である。これが可能であるためには、「パソコン」即ちパーソナルコンピュータが、その語義通りに「パーソナル」である必要がある。

結局、ここで求められるユーザの自由度と、前項の結論であった、管理の用意なシステム、とを両立すること、という厄介な課題が、ここから導

30) 特にこの傾向を統計的に調べた訳ではないが。

31) 筆者は能動的な学習行為を総称してこう呼んでいる。

かれることになる。

### 5.3. ユーザ層の多様化

先に述べた用途の多様化に加えて、ユーザ層の多様性も考慮しなければならない。これまで計機は特別な学部や学科（本学においては産業情報学科）のものであったが、今は、大学生の基本的な勉強（および研究）の道具でもある。さらに生涯学習など大学のオープン化の方向を考えると、対象ユーザの枠は、近隣の市民全体<sup>32)</sup>にまで広がる。従って、筆者は *NUANCE* を、教室、図書館の次に位置付けられるべき大学のインフラストラクチャとして捉え、その前提で設計している。

ユーザ層<sup>33)</sup>が多様だということは、情報システムに求めるもののレベルが多様だということである。情報のプロを志す者は、前節の「パソコン」との付合いを含めて奥深い勉強をしていく必要があり、その準備としてキーボード操作を始めとする「基礎」の習得のための時間も用意されている。一方、情報システムとの淡泊な交際を求めるユーザもいる（例えばレポートを清書したいだけ、など、一般にはこのユーザ層の方が圧倒的に厚い）。前者には、ホームポジションから指を離さない迅速な、後者には、「はい/いいえ」と「これ」を1本指で示すことにより、またその中間の人にはそれなりに中間的な、それぞれオペレーションの手段が提供される<sup>34)</sup>のが望ましい。

また、前者には、「パソコン」を提供し、機械とのつきあいも含めて学んでもらわなければならないのだが、一方、後者は、そのようなことを望んでいる訳ではない。「ディスク」を「初期化」し「システム」を「転送」して「ボリューム名」

をつけたり、定期的に「バックアップ」をとったり、はたまた「エディタ」で「CONFIG. SYS」を書き換えたり、こういうことがマシンの前に座る目的では決してないし、こういう呪文を覚えたとも思っていない<sup>35)</sup>。

即ち、大学のインフラとしての情報システムにおいては（社会全体のほぼ平均値といったところだろうが）多くの人にとって、コンピュータは、目的ではなく何等かの機能を提供してくれる「道具」であればいい訳である。繰り返しになるが、彼らの求めるものもまた、パソコンはパソコンでも Personal Computer でなく Personal Computing なのである。勿論（すでにお気付きだろうが）、*NUANCE* の 'C' も 'Computing' の 'C' である。ちなみに、計算機上での諸活動を研究する学問として、Computer Science（計算機科学）があるが、計算機が目的でなく道具であるケースが多くなるのに呼応して、Computing Science という用語も使われ始めているようである（これに相当する日本語の術語は、認知されたものとしては筆者は知らないのだが）。

その一方で、前述のように、パソコンをパソコンとして使い、つきあって行くことを学ぶ必要のあるユーザ層もいる訳で、つまり、「よく管理され整備された Computing 環境」と、「ユーザが手を加える余地」との両立という、結局、結論は前節とほぼ同じになる。

### 5.4. セキュリティ

一般にシステムの健全な運用を妨げる「不測の事態」とされているものをここに整理しておく（ここに予測している以上は、定義上「不測の事態」ではなく、これ以外の所に真の「不測の事態」があるのだろうが）。

#### 5.4.1. 故障

機械は故障するものである。当然起こりうるその事態のために、例えば授業が停止することがあ

32) そのすべてが日本人であるとは限らない。英語と日本語以外の言語（特にアジア諸言語の比率が高い）を使うユーザも想定しておかなければならない。

33) この言葉が人間そのものの上下関係とは無関係であることは言うまでもない。

34) 現実の製品はまだいづれも不十分である。さらには、ハンディキャップを持つ人のためのユーザインタフェースを準備することもこれからの重要な課題なのだが、*NUANCE* では残念ながらその取り組みすら始まっていない。その主たる部屋がエレベータのない建物の3階にあるという、その出発点から願っている状況である。

35) この部分はシステム提供の立場の者が考慮すべきこととして書いている。一方で、筆者はそれ以外の人（特に学生）に対しては、ブラックボックスをブラックボックスとして受容するのではなく構造やしくみに関心を持ち 'hack' してみる（およびそういう生活態度）を奨めている。

っては困る。大学授業一コマの価値（金額に換算するなら納付金を総時間数で割った額になるだろう、大変な額である）を考えると、例えばホスト機が故障したので補修部品が届くまで3日間実習授業ができない、というような事態は避けなければならない。100台のユーザ機のうち1台や2台の停止なら、「空いてる席に移る」という簡単な対策が可能であるが、システム全体が依存する基幹部分については注意が必要である。

なお、故障に起因するトラブルとしては、機能の停止以外に、情報の喪失という場合もある。ディスクのクラッシュ（これは決して珍しい事態ではない）の場合などの場合に生じる。部品を交換するだけでは回復しないものもあるということである。

#### 5.4.2. 誤操作

人間は過ちをおかすものである。大規模システムにおける不測の事態(事故等)の事後分析等で、「操作ミス」「ヒューマンエラー」で片付けられてしまうケースが散見されるが、本来これはシステムの欠陥である。ユーザの誤操作を当然あるものとしてシステムを作らなければならない。また、誤操作を恐れずに新しい試み果敢にアタックすることを奨励することは「公園型」環境の勘所でもある。

何か重要なファイルを消してしまうというのがよくあるケースである。ユーザの個人的なデータに関してこれを防ぐ方法はないが（ユーザの困惑を最少限に押さえるため、定期的バックアップを行うというサービスは有効だが）、システムにとって必要なファイルは、一般ユーザからガードされた領域におかれるべきだろう。このガードが不十分だと、例えばウイルスの侵害も受けやすい。また、情報の漏洩にも考慮する必要がある。情報システムはパーソナルなメディアでもあるから、個人の Private な情報は（その所有者が意図的に公開したものでなければ）、他人からアクセス可能であってはいけない。

誤操作の中には、知識不足や過失だけでなく、悪意に基づくものも存在する。ネットワークによる外部接続された部分に関しては、セキュリティ

は特に重要である。これについては、「防火壁」<sup>36)</sup>を設けるなどして対処する。内部からの侵害、例えばオープンスペースに設置された機械に対する破壊行為や窃盗行為等も、可能性としては存在するため、考慮しない訳にはいかない。使うための設備であるから、大切な機械だからといって倉庫に押し込めておく訳には勿論行かないが、ある程度の歯止めは必要であろう。

#### 5.4.3. 備えの程度

セキュリティの強さは、利用度や使い勝手と反比例すると言われる。機械や情報に対する自由なアクセスを侵害しない程度であるべきだし、また、完全なものを求めると非常にそのコストが高価なものになるため、予算や状況との兼ね合いで調整する必要がある。

システムの2重化という方法がある。システムの停止を防ぐために、基本的には有効な手法である。例えば、管理中枢となるマシンを2重にすると、一方がダウンしても運用が継続できる。この場合、大別すると筐体内での2重化と、ネットワーク上での図1(b)のような2重化の2通りが考えられる。すると、前者ではネットワークとの接続部が、後者ではネットワークそのものが、非多重化部となる。この非多重化部に障害が発生すると、システム全体が停止する。そこでネットワークを多重化する。ネットワークの場合はループを作る（ブリッジにより経路制御は必要）という「冗長化」により信頼性は向上できる。この辺までは、運用の安定性のために是非必要な対策である。

しかしこれでも完全とは言えない。2重化された両側の部分を接続し同期させたり、或は主従を切替えたりする機能が必要になるが、その部分に非多重化部分ができることは殆ど避けられない。或は、苦勞して（コストも払って）これを完全な姿に近づけたとしても、停電、災害といった、建物の外側で生じる障害の影響は免れない。かといって、生起確率の低い事象にまで対象を広げて完全を追求していくと、そのコストは莫大なものに

36) 物理的な壁ではなく、ネットワークにおいて外部と内部を隔てる壁として機能する論理的なネットワークである。



なる。どこかに妥協点を求めるしかない。

幸い、大学という所はかなりおおらかな所で、例えばシステムが15分間停止しても、銀行システムや列車制御システムのように社会的に影響を及ぼすことはない。この程度の停止は、むしろ「いい経験だ」と考えて積極的に（教材として）活用することすらできる。教育システムの場合、運用継続性に関しての社会的責任は（無為無策は許されないとしても）、比較的小さいと考えていい。従って、**NUANCE**では、システムのノンストップ性は追求せず、15分程度の停止は許容できるようにした。冗長に用意されたシステムは、ホットスタンバイでなく、コールドスタンバイさせておき、障害時には、管理者が来て、何本かのケーブルの抜き差しと、予備機の起動をする、この程度の処置（15分程度と見込まれる）で復旧可能であればいいということである。夜間や休日（入室管理の問題はあるとしても、少なくとも在宅アクセス等は24時間サービスは可能である）に生じたトラブルは、次の（誰かが来る）朝まで解決を待たなければいけないが、この程度はコストとの兼ね合いで許容するのが適当だろう。

実際には、これよりも厄介なのは、データの消失である。これは、メーカーのエンジニアにも直せないし、待っていれば復元するものではない。たとえ実習の過程で生じる出来損ないプログラムであっても、同じものが世界に2つとないかけがない情報資産であることに変わりはない。従って、すべてのデータ（キャッシュやスワップといった一時的なものは除く）が、障害時にも短時間で復元可能な冗長性を持っていること、という程度の条件は設ける必要があるだろう。勿論、データの緊急度の度合によって復元時間に差違がある（リムーバブルなメディアやダイレクトアクセスでないメディアに保管される等）のは許容していいだろう。

なお、上記の（装置およびデータに関する）条件は、いずれも一時に単一の障害の発生だけを想定する。同時に複数の障害が発生するという（確率論的には）希な事象までを想定した場合、その想定すべき事象の数は、2乗、3乗で増加し、その対策は技術的にもコスト的にも困難であるから。

最後に、ユーザの悪戯の類に関してであるが、これを防ぐ決定的な（ローコストかつ有効な）方法はない。実際、悪意があって、周到な準備さえすれば、何でもできてしまうものである。

**NUANCE**では、これについて基本的にはグブアップしながらも、善良なユーザの「つい出来心」を誘発することのないようにという、いわばキリスト教的な発想に基づいての条件は設定した。つまり、「(部品を) ちょっと引っ張って見たら、抜けてしまったので、丁度いいので持ち帰って使ってる」というような事態を避けるよう、ケーブルリング等に工夫を施す、というようなレベルの条件である。

## 5.5. 基本設計

### 5.5.1. 要求事項のまとめ

これまでに述べた要求事項を概ね以下のようなキーワードで要約することができる。

- ① 価格
- ② 安定性（運用継続性）
- ③ 安全性
- ④ 魅力（速度、使いやすさ）
- ⑤ 発展性
- ⑥ 集中管理
- ⑦ ユーザの自由度

特に **NUANCE** では、前述の相反する要求群を

- ・ユーザにとって：（どの席に座ってもそこに  
あるマシンが）私のマシン
- ・管理者にとって：（全部が）一台のマシン  
にそれぞれ見えること、という設計目標をたてた。

### 5.5.2. プラットフォームの選択

プラットフォームとは製品群の共通の土台のことである。システムのプラットフォームを選択するということは、そのプラットフォーム上に展開される（適用可能な）製品群を選択することである（その製品群の中から具体的にどれを選択するかはまた別の話である）。これまで、特に日本では図5(a)のようにハードウェアがプラットフォームとして機能していたため、プラットフォームの選択とは、計算機の機種（或はメーカー）選択とは

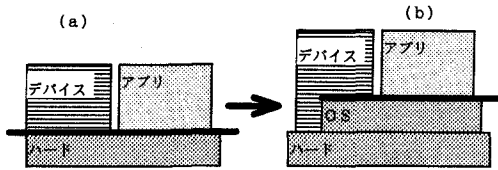


図5 プラットフォーム

ば同様であった。現在、市場は図5(b)の形に変わり、周辺機器等に関するハードウェアプラットフォームと、アプリケーションソフトウェア（以下「アプリ」と記す）等に関するソフトウェアプラットフォームとが存在している。以前はハードに合わせてソフトが作られていたが、ソフトに合わせてハードが作られるようになっており、また、ソフトが情報システムの主役ともなっている現在、まず決定すべきは後者のソフトウェアプラットフォームである。広義の（ブート機能やデバイス管理機能を自身で持たないウィンドウシステムを含む）OSの選択と言ってもいい。

これまで述べた議論を総合すると、現在市場にある具体的製品（または技術）の中から3者しか残らない<sup>37)</sup>。

- 1) Ms-Windows (+DOS) : 以下 Windows と書く
- 2) MacOS
- 3) X-window (+UNIX) : 以下「X」と書く

この中から1つを選べばいいのだが、できれば2つの並立が望ましい。プラットフォーム毎の得手不得手が存在し、また、様々なプラットフォームを経験する機会の提供もまた教育の場の責任であるからである。実際、本学の旧システムにおいては、不完全ながらも UNIX と DOS の2つのプラットフォームが用意されており、相互補完的に機能していた。この方針は継承するのが望ましい。

結局筆者としては、1)、3)を選んだ。2)の MacOS は、先進性、ユーザインタフェースの洗練度、特に芸術的分野での製品群の充実度などのメリットはあったが、下記のような点を考慮して、メインとしては1)が選択されることになった。

37) これ以外に NeXTstep も機能的に捨てがたい魅力はあったが、アプリの豊かさの点で候補に上げられなかった。

- 1)との並立は不可能 (Mac 上で Windows のエミュレーションという方向は可能ではあるが、まだ製品としては不完全なものしかない)。
- 1)2)共に OS は1社の所有であるが、1)では対応ハードがオープンな競争状態にあるのに対し、2)ではハードも1社の占有状態にある。
- その Mac は一体型の機器として供給される傾向が強い。特にユーザ機として適用可能な価格帯のものにその傾向があり、これは一般ユーザには魅力だが、陳腐化防止にはマイナスの要素である。
- 普及の度合や市場の広さにおいて、1)は2)に対して優れており、Mac用アプリの Windows への移植もかなり進行している。
- 後述のシステム構成による「パソコン」化の技術が見出せなかった。

ただし、仕様書においては2)を排除する記述は行っていないし、従って、機種を限定する条件も記述していない。ただ、機能や性能に関する条件を記述するためのモデル機として、Windows ベースの PC を前提にしたにすぎない（現実にそれ以外の提案は出てこなかったが）。また、3)については、

- エミュレーションにより1)や2)と共存が可能。
- ネットワークを利用するために UNIX の存在は不可欠。
- UNIX 上に (PC 文化とは別の) 優れた技術蓄積やフリーソフトという公共資産<sup>38)</sup>が存在する。

という点で、サブプラットフォームとして極力残すこととした。

### 5.5.3. ハードウェア構成

前節までに述べた議論の中で、システム形態の候補としては、

- ホスト=端末型
- スタンドアロンの PC 類

の形のもの（ここで詳しくは述べないが）す

38) 実際、教育機関でソフトの充実を図るためには、フリーソフトを活用しないでは困難である。

に、問題外として自動的に消えている。ユーザの前に置かれる WS としては、

- 1) EWS
- 2) X 端末
- 3) PC 類 (AT 互換機または Mac ; 本節の各項の検討は、実際には並行して行なっているため)

が考えられる (いずれもネットワークに接続されていることが前提となる)。

このうち、1)は、1つの機器としては最も魅力的な機器ではあるが、

- ・ユーザ機として100台設置するには高価。
- ・ユーザの前にノンプリエンティブ・マルチタスキングの、やや重い OS が呈示されることになり、その起動や停止にあたっての手順や条件などが、一般ユーザには負担になる可能性がある。

といった事情で望みが薄い。筆者が本誌上で以前に報告してきた<sup>[3,4]</sup>ように、ディスクレスやデータレスの運用等の工夫で、後者については解消の可能性はあるが、価格の問題は避けがたい。

その点、X 端末というソリューションは、価格の点でもさほど無理がなく、また、ユーザの前に置かれる機器は基本的にディスクレスであるため、情報に関するセキュリティの点で、心強い。問題は、

- ・Unix+X の上での Windows エミュレーションがまだ製品として不安定。
- ・5.3 節に述べたプロ指向の学習者にとって、物足りない点がある。

といった点である。

一方の PC においては、複数プラットフォームの提供という点で、

- ・Windows 上の X 端末エミュレーション製品が (たまたま *NUANCE* にとっていいタイミングで)、安定してきた (同時に併用が可能)。
- ・PC-UNIX もまた、安定し普及し始めた (起動の切替えにより併用が可能)。

という2つのソリューションが可能であり、勿論、価格や性能の点でもかなり優れている (ただし X 端末エミュレータはまだやや高価であるため、総合すると (本物の) X 端末と大差ないだろ

うが)。X 端末に対して1つ遜色となる点、次項に述べる共有における問題だということになる。これを解決できれば、PC が最も優れた解だということになる。

#### 5.5.4. 「パソコン」

一般に PC は、「パソコン」として製造され、販売されている。つまり個人が所有し使用することを前提とされている、個人のコンピュータである。このとき、機械と情報は同じ場所にあって構わない。一方、我々が求めるものは、個人のコンピューティングである。情報、および情報空間はユーザ各個人が所有しながらも、機械そのものは共有である。

個人の情報をどこに保管するか、これについて、従来は以下2つの選択肢があった。

- 1) 固定媒体：ハードディスクなど。
- 2) 可搬媒体：フロッピなど。

特にハードディスクは最近大容量化と低価格化が著しく、そのことだけを拠り所にして (即ち、この部分で節約する必要はない、などというトンチンカンなことを言って)、ローカルディスク、つまり各ユーザ機に設置したハードディスクによる運用形態に疑問を感じないシステム構築者が多い。しかし、このローカルディスクこそが、これまでに述べた諸問題の要因になっている。ローカルディスクに頼ったシステムは、セキュリティはゼロであると言っている。個人の情報は漏れ放題、ウイルスも入れ放題である。OS によってはプロテクトをかけることが可能であっても、少し技術があれば、生のデバイスをスキャンすれば、何でも見える。第一、機械Aで昨日作成し保管した文書が、今日は席の都合で機械Bを使っている時に、アクセスできない。こんな不便なことはない。

一方の可搬媒体では、この問題は存在しないし、フロッピが持つ容量や速度の問題は、MO や MD などの最近の媒体 (提案乱立の傾向はあるが) の採用によりかなり改善され得るが、別の根本的な問題がある。持ち運びが可能であるということは、持ち運ばなければならないということである。その媒体を例えば家に忘れてきた場合、ユーザは情報にアクセスできない。これはユーザの

ミスではない。当然起こり得ることを想定できていない、システムの欠陥なのである。

結局、両方の問題を解決する方法として、3) ネットワークディスクを用いる、これが今考えられる最適解である。情報と機械 (WS) の置き場所は物理的に分離する (ネットワークにより接続されればよい) こと、前者はオープンスペース、後者はクローズドスペースに置くこと、ということになる。この機能は、ネットワーク OS (通称 NOS) の類のものなら、サポートしている (というかそのために NOS が開発された) し、セキュリティに関する問題も (少なくとも情報漏洩については) ほぼ解決している。

ただ、今1つの (文字どおりの) ブートストラップ問題がある。ユーザ機をどこからブート (起動) するか、即ち、NOS のクライアント側ソフト (ユーザ機で動作する) を、どこからロードするかである。起動に必要なこれらの情報資源もまた、誤操作や悪意から守る必要があるし、場合によっては個人で管理したい場合もあるだろう。すると1)も2)も不可、3)も原理的に不可、である。

方法はある。ROM を使うことである。ブート ROM と呼ばれる、この ROM には、起動に必要なコードをネットワークからダウンロードするための、最低限のコードが含まれている。この ROM に入ったコードを実行している計算機からのリクエストに対して、応答できる NOS があればいい。これがあれば、PC のディスクレス運用が可能であり、これまで述べた諸問題はほぼ解決する。

これは EWS や X 端末では一般的に使われている方法である。一般には知られていないが、PC でも使用可能である。主要な NIC (ネットワークインタフェースカード; 日本ではネットワークボードとも呼ぶ) は、この ROM を装着するソケットが準備されているし<sup>39)</sup>、メジャーな NOS はこ

れをサポートしている<sup>40)</sup>。この、リモートブートと呼ばれる方法による運用が、とりあえず *NZIANCE* の最大の特徴ということになるだろう。

なお、これにより機能的には1)や2)に頼らずに、PC をディスクレス運用<sup>41)</sup>は可能なのだが、ローカルディスクは装備した。これは、パソコン OS である Windows や DOS がスワップやキャッシュのためにローカルディスクを活用して性能を確保するためであり、また、裏環境としての PC-UNIX 等、ローカルディスクを必要とする場面もまた存在するからである。従って (Windows による表環境については)、データレス運用 (オープンスペースに情報資源を置かない) が行われていることになる。

## 6. おわりに

設計の過程で沢山の事を考えた。そのため、予定のバイト数を越えて長い文章になってしまったことをお許し頂きたい。つい、設計者の「思い入れ」のようなものが、込められてしまう訳である。抽象的な内容が多かったため、これだけで *NZIANCE* を理解して頂くのは難しいだろう。すでに稼働を始めているので、ぜひ実際に操作して感じて頂きたい。なお、遠方の方のためには、Internet を通じて *NZIANCE* に関する情報を WWW など提供していく予定でいる。

*NZIANCE* 構築に関して痛切に感じたことは、ユーザにとっての福音だと言える昨今の状況の裏側に、大量生産の論理に由来するユーザの苦勞と不自由が隠れていることであった。本稿は、その大きな流れに対しての、筆者のせめてもの抵抗に関する報告であり考察である。

筆者は以前、計算機メーカーで働いていたことがある。つまり供給側(その最も上流にあたる)で、

39) 調査時点では、これを知る人は (国内では) 少なく、このソケットを見て「何だろう」と疑問に思ったところから、筆者の探索が始まった。

40) 日本では NetWare 3.12J からサポートされ始めたのが最初である。ちなみに、この機能が日本でアナウンスされたのは、94年4月末頃 (*NZIANCE* 仕様書をリリースした後!) であった。筆者としては冷や汗ものの冒険だった。

41) 本項執筆時点では、日本語システムに関係する2、3のファイルに関しての問題を残しており、全てのファイルをローカルディスクから消すことはできずにいる。

一般ユーザとの接点を殆ど持たずに研究を行っていた、いわば供給側の論理の中で育った人間である。その私がユーザ側の立場に立つ（といっても初心者ユーザの視点に立つことはほぼ不可能だが）ことにより、初めて見えて来たことも多い。

たまたま筆者の場合は縁あって今の職に移り、この両方の視点を結果的に経験することが出来た訳だが、職業間の流動性の低い日本の社会においては、こういう幸運に恵まれる人は少ないだろう。この流動性の低さこそが、多くの社会問題（双方の立場の主張の衝突）の解決を困難にする要因かも知れない、ということも考える。

ともあれ、筆者のこの経験は（今回<sup>42)</sup>については）貴重なものであり、幸運なものであった。この機会を与えて下さり、かつ支援して下さいの方々に感謝している。当時の計算機システム委員会委員長であった谷口教授と今年度委員長の佐藤助教授を始めとして、大学運営に関る方々からは、3.1章脚注に触れたような、様々な形での「組織としての配慮」という支援を頂いた。また、旧システム構築時に主役を演じた和田助教授からは、筆者が専らその旧システムに難癖をつける役回りであったにもかかわらず、先達としての様々な助言を頂いている。また、*NUANCE* の調査開始に先立って、近隣の高等教育機関のよしみで、長野高等専門学校および信州大学総合情報処理センターの方々から助言を頂いた。また、4.6.3項で述べた調査に協力頂いた各業者、D社およびM社（念のため社名は伏せておくが）の担当者諸氏には、特に（業務の一環とは言え）御苦勞頂いた割に、結果的には報われなかったことについて、恐縮している<sup>43)</sup>。これらの方々に（一括で恐縮だが）、感謝の念を申し述べる。

#### 参考文献

- [1] 長野大学「基礎教育用計算機設備94年度調達主要部分要求仕様書 1.0版」'94  
 [2] L. Wall and R. L. Wchwarts『Perl プログラミング』アスキー出版局'93

- [3] 平岡信之「キャンパスネットワークにおける資源の有効利用(I)」長野大学紀要第16巻4号'94  
 [4] 平岡信之「キャンパスネットワークにおける資源の有効利用(II)」長野大学紀要第16巻1・2号'94

(ひらおか のぶゆき 講師)  
 (1995. 1. 26 受理)

42) 一教員にとっては学内の煩わしい無給の雑用であり、次回までには然るべき担当職が用意されていて欲しいという願いは、やはりあるが。

43) 現在実際に担当してもらっている会社に対しては、いずれ機会を見て言及することにした。