

平成 5 年度

標準化動向の現状とユーザーの

問題意識

ユーザーサイドから見た標準化の問題点の検証

平成 6 年 3 月

標準化動向調査研究部会

〔目 次〕

研究部会メンバー

第1章

- 1. 1 はじめに
- 1. 2 研究活動の経緯

第2章 ユーザーインタフェース及び通信機能関連

- 2. 1 画面に関する問題点
- 2. 2 キーボードの標準化に関する現状の問題点と改善提案
- 2. 3 マウスの標準化について
- 2. 4 プリントインタフェースの標準化について
- 2. 5 日本語入力
- 2. 6 ネットワーク関連機能
- 2. 7 LAN、WAN
- 2. 8 ドキュメント・データ標準

第3章 相互接続性、相互運用性、可搬性関連

- 3. 1 TPモニタ
- 3. 2 データベース
- 3. 3 オペレーティングシステム(OS)
- 3. 4 分散処理
- 3. 5 ネットワーク管理、分散システム管理
- 3. 6 文字コードセット

第4章 方法論、見積り標準関連

- 4. 1 CASEについて
- 4. 2 オープンシステム(クライアント/サーバーシステム)開発方法論
- 4. 3 品質保証
- 4. 4 見積り
- 4. 5 契約

終わりに

第1章

1.1 はじめに

近年の情報技術の飛躍的な発展に伴い、オープン化やダウンサイジング化は、もはや時代の潮流となりつつある。このこと自体はユーザーにとって選択の幅が広がり、より良いシステムをより安く効率的に作り上げられるという利点のある反面、それに関わる「標準化問題」は非常に重要な意味を帯びてくる。今までメーカーのサービスに頼りきっていたユーザーにとって、自らが積極的に標準の問題に取り組む必要があるにも関わらず、現在のところユーザーの意識はあまり高いとは言いがたい状況にある。

一方、情報関連の標準化団体やメーカー・ベンダーは今後の問題解決に当たっては、実際にお金を払うユーザーの意向や意見を抜きには物事が考えられない状況に追い込まれつつある。

このような背景から、実務に根ざした実用的な標準化を目指すことを目的に委員会・研究部会を設置することは誠に意義深いことであり、情報化に際しての標準化問題を広く捉え、ユーザーの視点から調査・研究を行うこととした。これからは機会あるごとにユーザーの意見・要望を積極的に汲み上げ、提言として表明していくことが必要であると考えている。

平成5年度は、日頃から多くのユーザーが経験している身近な課題から取り組むとともに、研究成果を広くユーザーに公開し、標準化の重要性をさらに認識してもらうことをねらいとして、啓発活動を積極的に展開してきた。

1.2 研究活動の経緯

研究部会は、標準化推進委員会における活動方針の審議・決定を経て設置され、平成5年1月に活動を開始した。

委員会の審議過程においても、また、研究部会初期の議論でも特に対象を限定せず自由にブレインストーミング的発想から、問題になっている、もしくは注目している項目を列挙してみたところ、多岐にわたる分野で“標準化”問題が表面化していることが改めて確認された。その結果、平成5年度の目標に関しては、「標準化を必要としている項目の“マップ”的なものを作成し、ユーザーがどういう項目に関心があるのかを調査する」ことになった。

標準化の動向調査をするに当たっては、当初、国際標準化団体の動向を知ることから始め、X/Open、UNIX International、Open System Foundationの各団体から、当年度の活動方針、考え方等についてプレゼンテーションを受けた。その後、関心の高い項目から、ユ

ユーザーの視点で課題や要望をまとめるため、User Interface 関係の問題点の列挙から活動に入ることにした。

平成5年7月、研究部会の活動を会員に公開し、より広く意見を収集するべく公開ゼミナール「標準化動向オープンフォーラム」の開始を決定。調査活動と並行して年度内に延べ4回開催している。

8月の部会からは第2ステップに入り、API、方法論の調査を行うことになった。

以上のような経緯で活動は推移してきたが、報告書作成に当たっては、あくまでユーザーの視点から各項目の現状と要望を表すことにした。この報告書は、現状を把握するためのハンドブック的なまとめを目指している。この報告書を読む方々に参考になれば幸いである。

第2章 ユーザーインターフェース及び通信機能関連

2.1 画面に関する問題点

検討作業を進めるにあたって、

- ① LOOK&FEELに注力する
- ② GUIアプリ作成時の手助けになるような物をアウトプットする

の2点を考え方の基本とした。

ハード的には画面の解像度、サイズなどの問題やEmulatorの問題などが考えられるが、現実的には何とか対応できているのではないかとということで、上記2点に問題点を絞り込むことにした。

また作業は図2-1-1のように進めた。

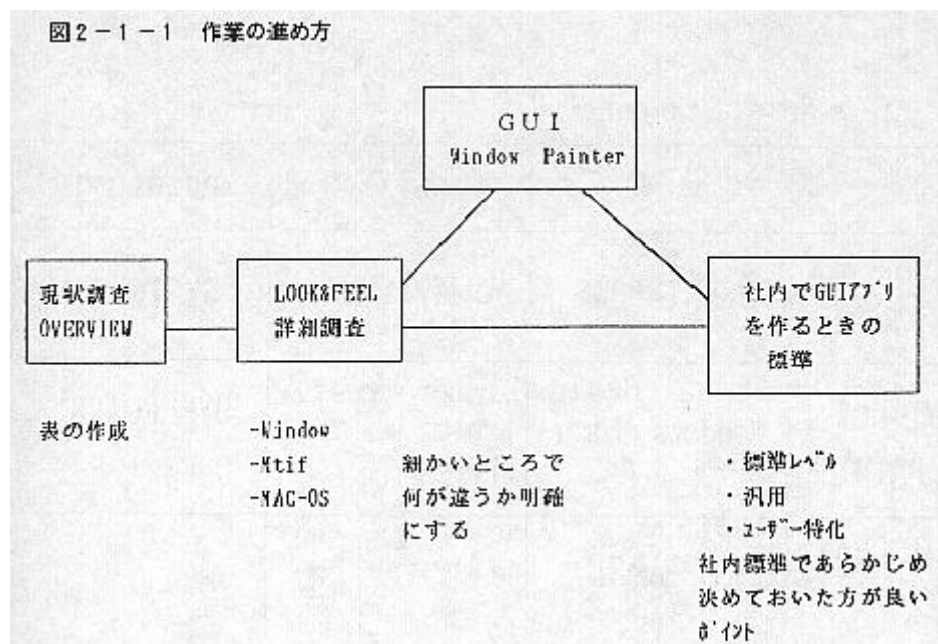


図2-1-1

(1)問題点(及び現状の認識)

- ① 全般的問題(主にPC、UNIX環境)

- ・ 画面解像度の基準がない

図2-1-2のとおりディスプレイはメーカー、機種により種々のフォントがあり、画面を全部表示できないなどの問題がある。

図2-1-2 各OSごとのUIの比較

	P C	UNIX	Macintosh
Look & Feel	MS-WINDOWS	Motif, OPENLOOK	SYSTEM7
WINDOW PAINTER	Visual Basic Windows Maker OPEN I/F	Builder Accessary INTERFACE ARCHITECT OPEN I/F	HYPERCARD OPEN I/F
DISPLAY	NEC 640×400 1.120×750 DEC SVGA 640×480 800×600 1.024×768 Quater L 1.280×1.024 AX CGA 320×200 EGA 640×200 320×200 640×200 320×350 640×350 VGA 720×350 360×400 720×400 640×480	DEC 3000 1.024×786 1.280×1.024 SUN SPARC10 1.152×900 1.280×1.024 classic 1.024×768 YHP 700 1.280×1.024 1.024×768 Fujitsu DS 1.280×1.024	640×400 640×480 512×342 1.152×862 1.024×768

図2-1-2

- ・ インタフェースビルダーのライブラリー等の互換性がない

[例]現状ではマシンにデペンドしたライブラリー、インクルードファイルが存在するため、

HP<-->RS (IBM)間のプログラム互換性がない
 HP800シリーズ<-->HP700シリーズのプログラム互換性がない

- ・ どういう商品を選択したらよいか、判断基準がない :

将来性のあるもの
 主流になるもの
 ローカルでないもの

- ・ LOOK&FEEL

MacはUIとして使えるか

OSの信頼性が低く、メモリー不足によるエラーが多い

Macのツール(Hypercard等)では、Windowタイプな画面が難しい

- ・ インタフェースビルダー

OPEN INTERFACEは本当にオープンな環境で使用でき、将来性もあるのか等

- ・ 画面設計上の問題

- ・ スタイルガイドの位置付け:標準として準拠するべきだと思うが(社内標準化での取舍選択になる?)

- ・ 画面展開は社内標準だけでよいか

画面関連図の記述方法(マニュアル、仕様書)

前画面、サブウィンドウの残し方

一度に開ける画面数

画面サイズ

リソースとの関連で標準を決める必要がある

- ・ 画面作成ツールの互換性

インタフェースビルダー、4GL、CASEツールの互換性は全く期待できない

- ・ PC(Window's)とUNIX(Motif)間でGUIソフトのPortabilityが乏しい。また、Windowソフトの考え方も異なる。

Look&Feelは極めて似ているが、細かいところでは異なる点もある。

今後クライアントマシンとして、UNIX、Windowsの両方が使われるような局面も当分は考えられ、APIの統一は重要である。

Window NT、Utlixware、SVR4. 2等、選択肢は多く、標準化動向の読みも難しく、ユーザー側のリスク、負荷が大きい

② PC系の問題

- GUIのLook&FeelはWindowsで統一されている点は評価できる
- Windows3. 1の普及期を迎え、DOS/VとNEC系の違いが少なくなっている
- NEC系とDOS/V系で解像度が異なる

③ UNIX系の問題

- GUIは2つの標準が存在するがCOSEに見られるように、Motifで統一する方向が明確になった
- X-Windowは標準ソフトになったが、C/Sモデルをサポートしている点で高く評価できる

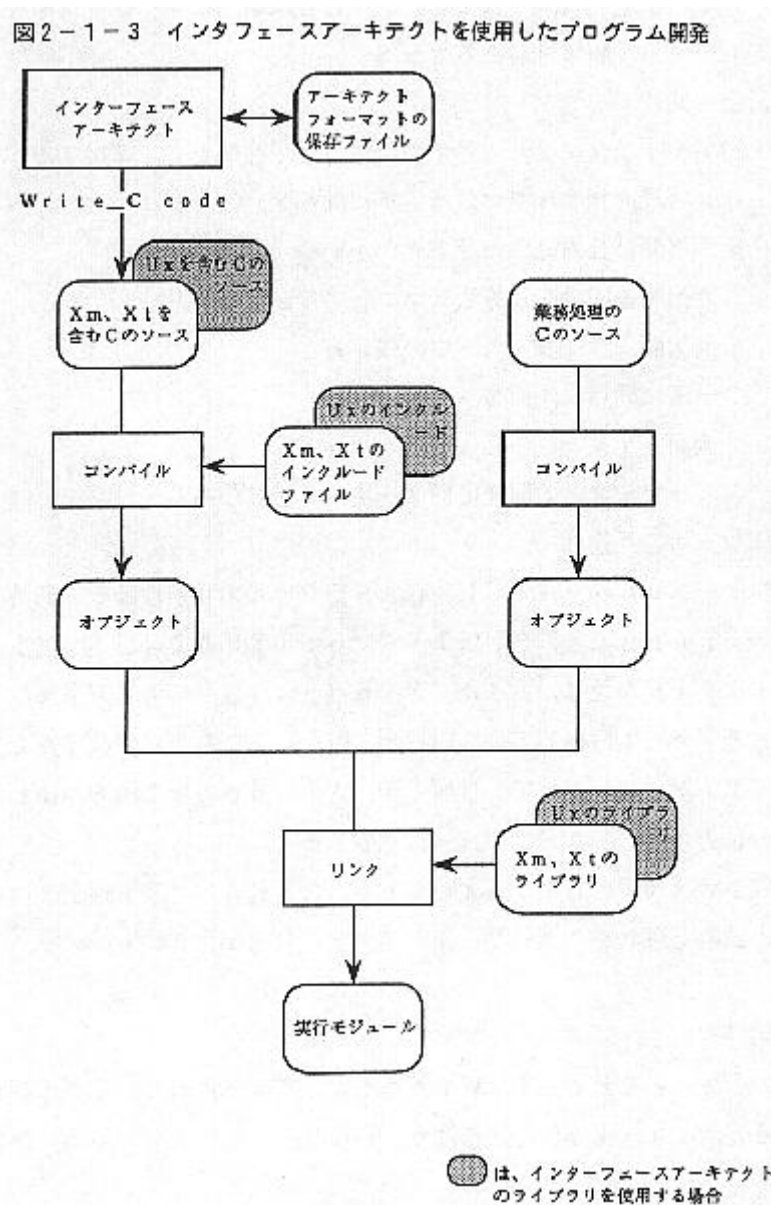


図2-1-3 インタフェースアーキテクトを使用したプログラム開発

(2) ユーザーから見て本来あるべき姿

- ・クライアントマシンのOSレベルを意識せずに、アプリケーションパッケージ、自作アプリケーションが使えること
- ・ユーザーは性能、コストパフォーマンスのみで、自由にクライアントマシンが選べること
- ・Look&Feelは細かいレベルも含め、OS間で統一されていること
- ・(Open Interface)はどこでも使える:リゾリューションも合っている)
- ・画面(UI)はユーザー自身で使いやすいように作るのがベスト

そのために、

- *ユーザーが作れるような簡便なツール提供と画面作成の標準化ができていること
- *プログラムは画面処理、業務処理、データベース処理と明確に切り分けされ、クライアント/サーバーモデルになっていること
(業務処理、データベース処理は情報処理屋が担当し、部品化が徹底されていること)

(3) 標準化進展の度合い

- ・ OSごとには、Look&Feelの標準化はほぼ決着している。
COSE-->APIの統一
- ・ 将来的にはマウス、キーボード、アイコンというインタフェースからVR的なものに発展する
- ・ PEN COMPUTERのような要素も重要になる
- ・ OS間の標準化作業は、OSの生き残り競争を考えると非常に困難と思われる
- ・ OSF Motif スタイルガイドを標準化とみなせば、メーカーによる差はなく、Motifでは標準化されている

*HP9000、RS6000のスタイルガイドはHP-UX、AIXとハード名が異なるのみで他は全く同じである

(4) 社内標準作成等への転用

Motif等のスタイルガイドはGUIアプリケーション設計上有効であるが、社内標準として使うには一般的すぎるので、スタイルガイドに準拠した社内標準を作成するという基本方針を提示する。準拠できないとか不都合な点があれば、スタイルガイドを改訂する(標準を決めることが重要)。

より使いやすいI/Fを作成するにはMotif等のスタイルガイドよりも細かい設計標準を社内標準として決める必要があり、その項目としては次のようなものが考えられる。

- ・ Windowのサイズ
- ・ 表示できるwindow数
- ・ オーバーラップウインドウの位置
- ・ タイトルバー
- ・ コントロールボックスの使い方
- ・ 最小化／最大化ボタンの操作
- ・ ショートセットキー
- ・ 画面関連図の記述方法
- ・ 用語標準
- ・ メッセージ

(5)ベンダー各社への要望事項

よりよいGUIのあり方についての標準化を推進するには、各ベンダー企業にその役割を期待することは難しい。OSを問わず、使いやすいGUIについて定義する標準化団体があった方が望ましい。

- ・ 日本語を意識したGUIのスタイルガイドを作成して欲しい。

例えば訳語の統一

- * MODAL、MODELESS、COPY、PASTE……etc
- * 日本語FEPの操作方法の統一 (GUI上での:非常にわかりにくい)
- * フォント:種類が多すぎる、外字フォントも統一が望ましい
- * 解像度:一定レベル以上のものしか作らないようにして欲しい

2.2 キーボードの標準化に関する現状の問題点と改善提案

(1)キーボードの経緯と現状

現在、各メーカーが出荷しているキーボードの起源は英文タイプライターから成り立っていることはご承知かと思われるが、キーボード内のキーの配列の設定経緯について若干触れておくこととする。

1876年にSholesが入力効率を考慮した「QWERTY配列」を考案し、その後1936年には、より一層入力効率のよい「DVORAK配列」を提唱したが、現在では「QWERTY配列」の方が一般的に普及している模様である。

昨今、コンピュータへのデータ入力装置としてのキーボードは「QWERTY配列」が主流となっており、JISハードウェア規格としては「QWERTY配列」に仮名文字を追加した「情報処理系鍵盤配列(X-6002)」を1980年に制定している。その後 1986年には仮名文字入力の打鍵の効率を改善した「仮名漢字変換形日本文入力装置用鍵盤配列(X-6004)」を制定しているが、各メーカーとも X6002を採用したキーボードが一般的に普及しているのが現状である。

次に、JISハードウェア規格の「QWERTY配列」とはどのような配列かを、JISハンドブックより引用し掲載する(図2-2-1)。

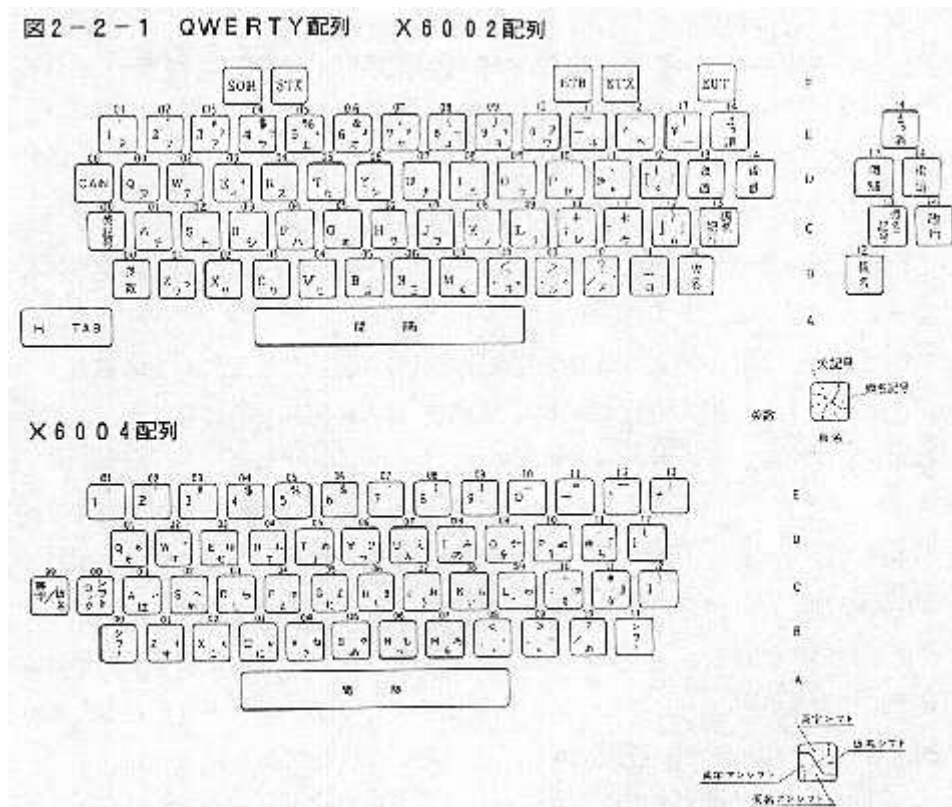


図2-2-1

ところで、現在一般的に普及しているキーボードの配列は、図形文字キー(英字、仮名、数字、特殊文字)はJIS-X6002に準拠しているが、制御文字キー、シフト機能キー、(プログラム)ファンクションキーなど、キーの機能や入力効率を向上させるための制御キー等が増加しており、それぞれのキーの位置や配列及び個数については各メーカーとも各様であり、さらには同一メーカー内でも、PC(パソコン)やWS(ワークステーション)ではファンクションキーの個数や制御キーの位置等が異なっているものも出荷されているのが現実である。

現在のデータ入力には各種業界に対応した多種多様な方式があり、各メーカーはユーザーニーズに合わせたキーボードも多種出荷しており、またユーザーニーズによる小型化へ

の対応としてノートブックタイプ型のような液晶ディスプレイ(液晶VDT)一体となった キーボードも普及してきており、様々なキーボードが氾濫しているのが現実である。

(2) キーボードの標準化動向

キーボードの国内及び国際での標準化推進としては、次の機関が活動を続けている。

- ・ 国際標準:ISO/IEC JTC1 SC18/WG9
- ・ 国内標準:JIS、日本事務機械工業会、電子工業振興協会、ANSI
- ・ 業界標準:日本語コンソーシアム(親指シフトキーボード)

ただし、上記機関においての標準化推進という点については、それほど主だった成果は上げられていないように思われるが、むしろ使いやすさを追求したキーボードの普及に注力が注がれているようである。これは、従前「日本工業標準調査会」でも建議されたように、

- ・ 「国内規格中心主義から国際規格基本主義へ」
- ・ 「メーカーのための規格からユーザー・消費者のための規格へ」

というように、オープンシステム指向への方向性が打ち出されているからだと考える。

標準化動向調査部会としては、オープンシステム指向との関連も鑑み、キーボードの操作性や選択性という点に着目しながら標準化提案を検討してきたものである。

(3) キーボードの問題点

現在、一般的に普及しているキーボードの操作性及び選択性についての問題点を追求していくと、次の点が上げられるのではないだろうか。

- 1) 現在の一般的なキーボードは入力効率の向上を狙いとして作られたキーボードであるが故、キーの配列 が不規則でありコンピュータに慣れ親しんだ人には、それほど入力しづらいものではないが、初心者にとっては使用しにくいものではないだろうか。
- 2) 熟練者でも、制御キー、シフト機能キー等の配列が異なる各社のキーボードを使用する場合には、キー位置の確認に手間取ったり、キー押下の誤操作の危険性が高くなり、操作性という点では劣化しているのではないだろうか。
- 3) 現在のキーボードはVDTと一体化しており、VDTを入れ替えるとキーボードも変更せざるを得ないよう、メーカーの戦略に嵌りユーザーの選択の余地がなくなっているのではないだろうか(ノートブックタイプ型 も同様なことが言えるのでは……)

これらの問題点の背景としては、各メーカーやベンダーが独自の特色を生かすための企業戦略や、ユーザー企業からの要望等により発生した問題ではないかと考えられる。現状これらの問題点に対し、標準化を含めた統一性を全面に打ち出すことは困難なことかと思われるが、標準化への第一歩としての指針を次に示したい。

(4)キーボードの標準化要望

「オープンシステムにおける操作性」の定義とは、『どのコンピュータでも自分に合った好みや操作方法で使用できること』が提案されている。これはキーボードやマウス等、各種の入力機器を対象とした広義の操作性として捉え、標準化部会としてはキーボードのみに焦点を当て、標準化提案をすることとしたい。

① キーボードの操作性に対する基本的考え方

1) エンドユーザーから見た操作性

初心者や熟練者を問わず、操作する側にとって最も使いやすく、かつ使い慣れたキーボードができること。

(例えば、JIS配列と五十音配列のキーボード選択など)

2) 開発者から見た操作性

開発者にとっては、メーカーあるいはベンダーに依存しない標準的な操作方法が規定されるか、または開発者が選択できるような操作性に関する標準仕様が組み込まれた標準APIを規定し、独立させることが必要であること。

3) システム管理者から見た操作性

マルチベンダー環境化での端末操作性を考慮すると、特に操作性を支援するソフトウェアやミドルウェアに関しては、ベンダーにかかわらず統一されていることが望ましいこと。

以上の点を勘案し、標準化に対する4つの提案をしたい。

② キーボードの標準化に対する提案事項

1) 使い慣れたキーボードが新規購入VDTにも接続でき、また他メーカーのVDTにも接続できるよう、ハードウェアの接続性を確保すること。

2) 制御文字、シフト機能キーの配列が使い慣れた操作位置に変更でき、さらに機能の変更及び追加が容易にできるようにすること。

3) 初心者や必ずしも高速入力が必要でない人には、JIS配列でなく正配列(ABC～、アイウエオ～、順)の方がキーボードへの抵抗が少なく慣れも早くなることより、情報処理専任者用キーボードはより入力効率のよい方法を目指し、初心者用キーボードはより使い勝手のよい方法を目指すべきであること。

4) コンピュータ教育の見地に立った場合でも、情報処理専任者教育と初等教育とを区別し、初等教育では正配列キーボード(「あいうえお表」に準拠した、右端上部から下方へのキー配列となった初等教育キーボード)等の、よりユーザーフレンドリーなキーボードが望ましいこと。

以上これらは、現状キーパンチャーや情報処理専任者以外の一般の人が直接データを入力する機会が増加してきていることや、情報処理教育の場が中等・初等教育の中に取り込まれてきている傾向があることより、それぞれの技量やレベルに合わせた対応が必要であるとの考え方から成り立っているものである。

以上の提案点をまとめると、図2-2-2に示すような「標準インターフェース」が必要になるものと考える。

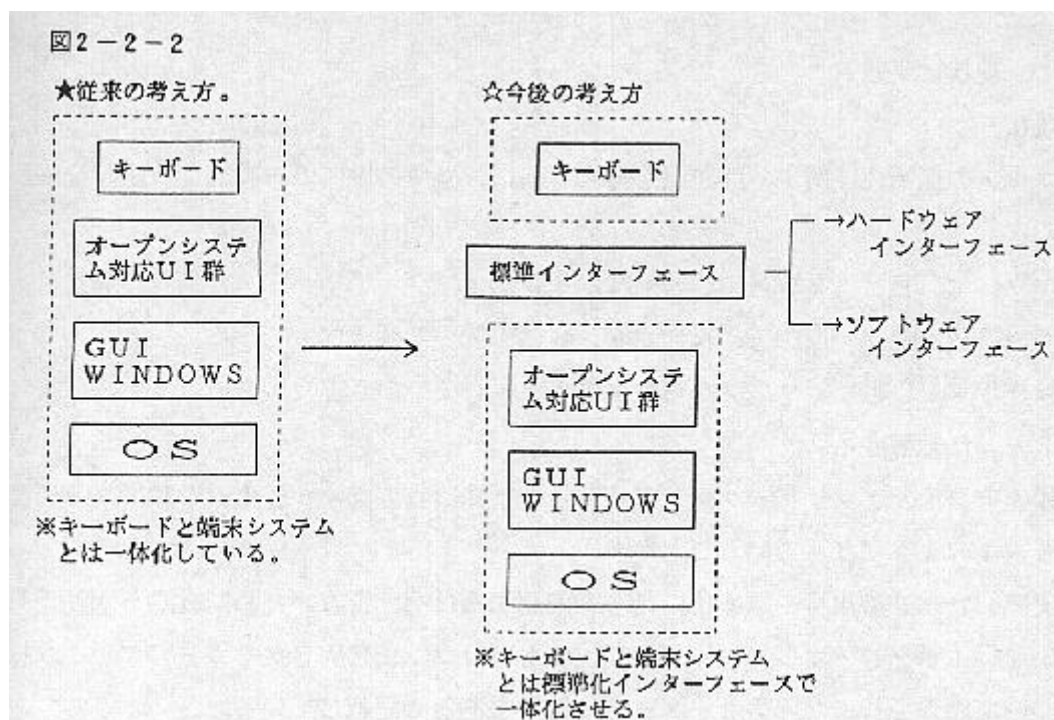


図2-2-2

ハードウェアインターフェースについては、物理的な位置や寸法、コネクタ形状、電圧・電流、プロトコル等を各メーカーが合わせる必要があり、いわばOSIと同様な取り決めを行い各メーカーがそれに準拠することが現実の第一歩である。

ソフトウェアインタフェースについては、ユーザーオリエンテッドに対応するキーカスタマイズ機能を仕様として組み込む等が必要であることは明かであり、現在でも各メーカーでサポートしている「キーバインディング」と組み合わせたソフトウェアインタフェースの作成が急務であると考ええる。

ハードウェアインタフェース、ソフトウェアインタフェースとも先の図に示す「標準インタフェース」に組み込み、早急を実現させることが標準化への第一歩であることは間違いないものと考ええる。

③ キーボードの標準化に対する提案事項達成後のメリット

上記の提案事項が達成されることによるユーザーメリットは、次の結果がえられるものと考ええる。

- ・使い慣れたキーボードを新規端末購入時にも流用でき、経費削減の一環となる。
- ・操作性や入力効率が向上し、入力データの品質も高くなる。
- ・古いキーボードでも、機能をバージョンアップすることができる。
- ・コンピュータ活用の量的・質的拡大が図られ、情報化社会の進展に寄与する。

(5) キーボードの標準化要望のまとめ

昨今、PC-LAN等のクライアント／サーバー・システム(CSS)が世の中に普及されつつある現状で、特にクライアント側はマルチベンダー環境が可能となっている中、VDTとキーボードのハードウェアが選択できないのは、本来のオープンシステム化とは相反しているのではないだろうか。またCSS内にはよりよいソフトウェアやミドルウェアが取り込まれていく過程において、それぞれのソフトの操作性が異なれば、エンドユーザーにとっての操作性劣化が顕著に表れることも明らかであり、この点も本来のオープンシステム化への支障とも考えられるのではないだろうか。

CSSにおいて、クライアント側は本来エンドユーザーが操作すべき対象のハードウェアと考えれば、この部分についてもソフトウェアやミドルウェアをも含めた操作性・選択性という点で、標準化指向とを鑑みたオープンシステム化が取り入れられるべきものと考え、今後の標準化進展に期待したい。

2.3 マウスの標準化について

(1) 現状と問題点

① 現状

マウスの標準化に関する問題点を考える前に、マウス操作の一般的な仕組みを確認する。

マウスは、PC、EWSの代表的なポインティング・デバイスであり、形状はボール組み込み型が一般的で、操作ボタン数は1個～3個となっている。

操作ボタンの数が、マウスからマウスドライバまでの処理(図2-3-1の〈1〉〈2〉)に影響する。

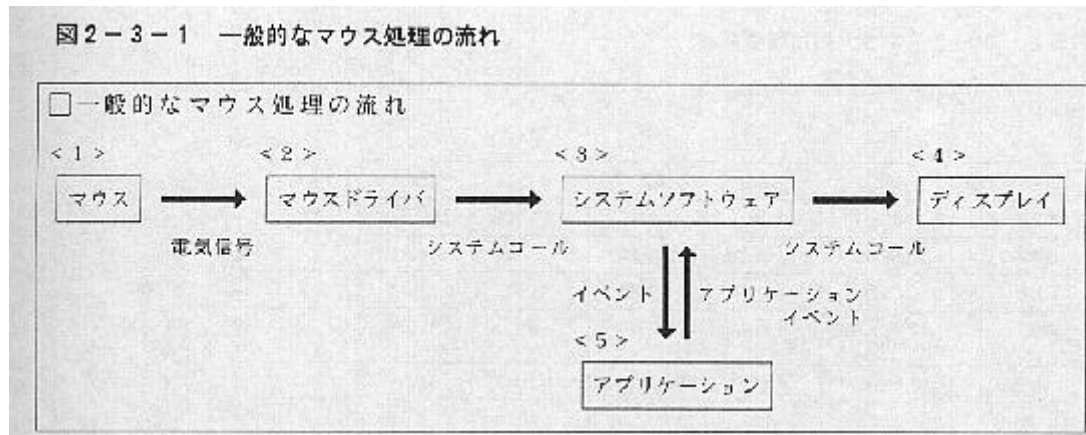


図2-3-1

操作方法は、アプリケーションに依存し一概には言えないが、ボタン数により表2-3-1のようになっていることが多い。

表2-3-1

表2-3-1 ボタン操作と代表的なコンピュータ

ボタン数	1ボタン	2ボタン	3ボタン
機種	Macintosh	パソコン	EWS
選択	1回クリック	左	左または中
確定	〃	左	〃
起動	2回クリック	左(2回)	左(2回)
解除	再クリック 対象範囲外をクリック	右	右 対象範囲外をクリック

実際、マウスの規格や標準化の動きを確認するため、特許のためMACしか使用していない1ボタン・マウスを除外し、主要なパソコン、EWSメーカーのマウス仕様について調査を行った。表2-3-2に結果をまとめ記載している。

表2-3-2
(ファイルが見つからない)

マウス処理の違いを、操作とソフトウェアとの関連でみると図2-3-2のようになる。

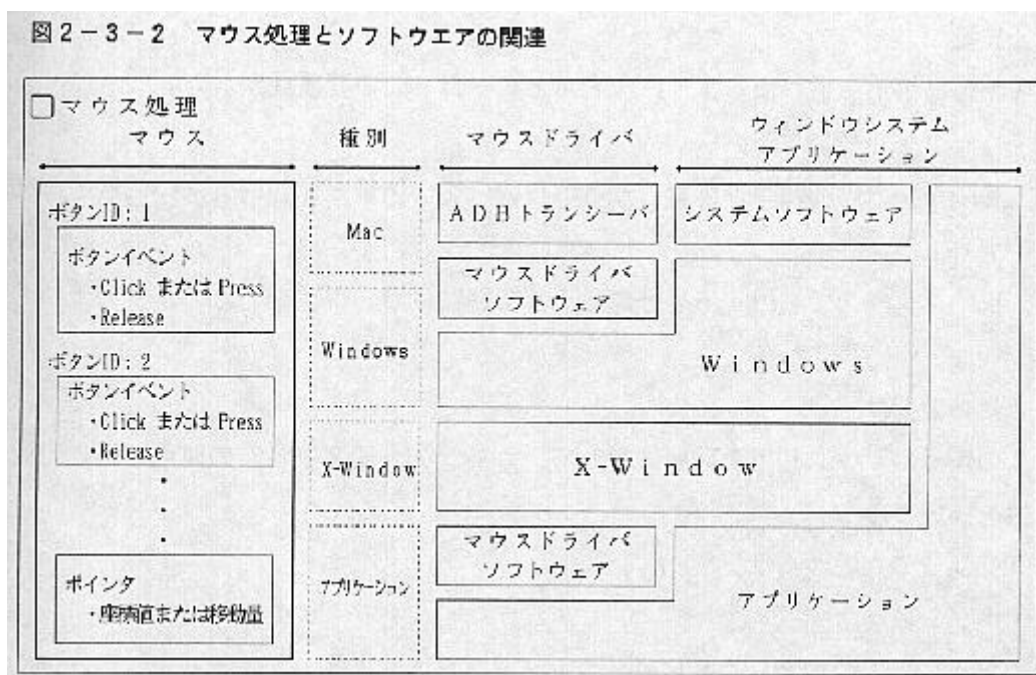


図2-3-2

② 問題点

調査結果や部会メンバーの意見から、以下のようなマウス利用の問題点が指摘される。

1) ボタン数

- ・ 最近は業務により異なる機種を使い分けることも多くなっており、ボタン数の違うMAC、パソコン、EWSごとに操作を切り換える必要がある。利用台数の多いパソコンは同一メーカーであってもボタン数が異なる例もある。
- ・ MACで使用する1ボタン方式では、初心者にとってダブルクリックを間違えやすく、慣れるまで時間を要す。

・ EWSでの使用が多い3ボタン方式は狭いところに押し込められているようで使いづらく、中ボタンが利用されることが少ないので、3つは多すぎるのではないか。

2) 操作法

・ボタン操作はアプリケーションに依存しており、操作法が統一されていないため各アプリケーションごとに
修得し、切り換えて使わねばならない。

3) 装置

- ・調査結果からもわかるようにほとんどメーカー間の互換性がない。
- ・コードが邪魔になる。
- ・移動量が自動的にコントロールされない(微小領域/比較的大きい領域の自動識別)
- ・狭いスペースで利用する場合にボールで移動させる現状の方式のままでよいのか。

(2) 必要要件

このようなマウス利用の問題に対して、ユーザー、開発者各々の観点から理想的な形態を描いてみる。

*ユーザーの立場からは、

① ボタン数

・ MACの1つボタン、PCの2つボタン、EWSに多い3つボタンに各々の文化があるが、YES/NOが明確な2つボタンが好ましいように思われる。

② 操作性(修得と作業効率)

- ・アプリケーションの種類によらず、少なくとも同一機種ではボタン操作が同じ意味を持つことが望ましい。
- ・マウスのボタンに解釈を任せず、アプリケーション側でアイコンやメニューの指示により統一された処理ができれば操作がずっと楽になる。実際、コンピュータ本体の性能アップもありマウスの制御自体もアプリケーション側が行うものも出始めている。

*開発者の立場からみると、

③ 移植性の確保

- ・ マウスに関するアプリケーション開発用ライブラリが標準に沿っている、またはマシン間での変換ライブラリがある等で移植性が確保されている。
- ・ マウス以外のポインティング・デバイスも同様に扱える。

(3) 標準化の動向

① 装置

- ・ 装置規格については公的な標準化機関の規格は見あたらない。
- ・ 1ボタン方式については、アップル社が特許を保有しており、操作法は規定されている。
- ・ パソコンで大半を占める2ボタン方式については、海外ではPS/2仕様がdefacto標準となっているが、国内では市場占有率第1位のNECが独自仕様を採用しており、当分は統一されにくいであろう。
- ・ EWSで使用されている3ボタン方式については、主要メーカー(サン、IBM、HP、SGI)で仕様が統一されていない。

② ソフト

- ・ 業界標準に依存している。Windows, X-Window (OSF/Motif, OLIT)では、標準的なボタン数、ボタン操作の解釈が規定されているが、カスタマイズが可能となっているためアプリケーション間での操作性の統一は無理である。
- ・ 現状のMAC、PC (Windows)、EWS (UNIX)の棲み分けは当面変わりそうもないが、パソコンでのWindowsの普及はめざましく、ワークステーションを見据えた上位版のWindows/NTの普及次第で勢力範囲が変化し2ボタン方式が事実上の標準になる可能性がある。

(4) 社内標準作成時の注意事項

このようにマウス規格、操作法についてはメーカー間、アプリケーション間でまちまちなため、現状では機種、アプリケーションを限定しないかぎり統一操作環境の実現は見込めない。要は、当面は違いがあることを前提に環境整備を考えることである。

自社開発の場合には、パソコンのWindows、EWSでのX-Windowsのカスタマイズ機能を活用すれば、ほぼ統一された操作環境が実現可能となるが、二重開発の問題が残る。

(5) 要望

ここまでマウスに限定して検討してきたが、パーソナルなコンピュータ利用のポインティング・デバイスとして、現状のマウスが必ずしも最善とは言えず、範囲を広げてメーカーへの要望事項を考えてみたい。

①装置

- ・ ボタン数に係わりなく共通仕様とし、機種間の互換性を持たせるべきである。
- ・ キーボードも同様であるが邪魔なコードがないコードレス・マウスがよい。
- ・ マウスの置き場所(マウスベッド)をコンピュータ本体のどこかに確保する。
- ・ ボールが汚れやすいので、直接外部に触れない装置方式を望む。

②操作

- ・ これだけ装置製造技術が進歩しているのであるから画一的な処理はやめ、微小／大領域に対応した柔軟な移動操作量の自動認識機能くらいはそれほどコストをかけずに付加できるのではないか。
- ・ 狭い机上スペースでも操作可能にする。
- ・ コンピュータの管面が直立状態でマウス操作場所と切り離されているが、管面の方向／位置やサイズの縮小等により、直接ポインティング可能な装置オプションも追加したい。

③ソフト

- ・ 同一ソースコードで機種やマウスのボタン数が異なる場合でも対応可能な標準化されたマウス操作の共通インタフェース(開発用ライブラリ)の提供。

④強力な標準化機関の推進

- ・ マウスにメーカーごとの独自性を持ち込んでも大してメリットがあるとは思われない。ユーザーを混乱させるだけであり、ハード仕様統一のための推進機関を定めるべきである。
- ・ マウスは、今後ほとんどがウィンドウ環境下で利用されると考えられるので、Windows 陣営とX-Window 陣営の接点を見いだすために、メーカー間も排除から協調へ転換し、ユーザーの立場から米国の主要ベンダーへの強い働きかけを行ってほしい。

2.4 プリンタインタフェースの標準化について

(1)目的

オープン化・マルチベンダー環境が急速に進むなか、各メーカーの特色を活かした高性能プリンタが提供されてきている。ビジネス・システムにおいて、帳票をきれいに早く正しく印刷することは必須の要件である。

しかしながら、各種プリンタを自由に使いこなす上において、幾多の問題点が見られ、オープン化・マルチベンダー環境における、利用者にフレンドリーなビジネス・システム構築への一つの障壁となっている。

- ・ 利用者は、異なるプリンタを同じ操作で共用することができる
- ・ 提供者は、個別にミドルソフトを考える必要がなくなる
- ・ アプリケーション・プログラム作成者は、プリンタの機種を意識する必要がなくなる

これらを実現できる標準化を進めたい。

(2) 現状と標準化動向

① ハードウェア・インタフェース

表2-4-1参照。

表2-4-1

インタフェース	RS-232C	セントロニクス	GP-IB	SCSI
インタフェース種類	シリアル、双方向	シリアル、(双方向)	パラレル、双方向	パラレル、双方向
マスター/スレーブ数	1:1	1:1	1:N	1:N
最大ケーブル長	数10m	数m	20m	6m, 25m
最大転送レート	~20Kビット/秒	250Kバイト/秒	1Mバイト/秒	1.5, 4Mバイト/秒
ハード的な普遍性	かなり高い。コネクタの形状、ピン配置に注意	高くない。信号のピン配置が各社まちまち	高い。形状についても規格あり	高い。ケーブルの供給インポートに注意
特徴	簡単・随時に接続できる。低速。エラーに対するリカバリができない	PCでは標準的。WSでは装備していないものあり。双方向は一部の機種	PC, WS共オプションでサポート。計測機との接続で作られたものがベース	WSでは標準。各種周辺機器に対応した標準コマンドを想定している

・ 国際標準・業界標準の動向

IEEE1284 D高速双方向パラレル・インタフェースの標準化の動きあり。HPとマイクロソフトが進めているECP(セントロニクスの上位互換)と、IBM・インテルが進めているEPPという仕様に割れている。

SCSIでは、8ビットから16/32ビットに拡張したSCSI-2がANSIで標準化された。

・ インタフェースの選定基準

基本的には、プリンタ側が持っているインタフェースに合わせなければならないが、転送速度から見れば、

*印字するデータがキャラクタのみで大量に印字を行わなければ

RS-232Cやセントロニクス

*イメージデータを印字する場合や大量にデータを印字する場合には

SCSIやGP-IB

データ転送速度の他に、コンピュータとプリンタ間の双方向通信を考慮する必要がある。プリンタの内蔵フォントなどのリソース情報や、プリンタの状態をコンピュータに取り込み、プリンタをきめ細かく制御するためには双方向通信が必要。

② ソフトウェア・インタフェース

- ・ 主な制御体系 表2-4-2参照。

表2-4-2

表 2 - 4 - 2 主な制御体系

I D M	5577, 5587, AFP	C A N O N	LIPSⅢ
E P S O N	ESC/P, ESC/P [®] , ESC/PAGE	京セラ	日本語 [®] ディスプレイ [®] プラス
A D O B E	POST SCRIPT	ブラザー	BR-SCRIPT
H P	PCL L/4, L/5	リコー	RPDL
N E C	PR201, NPDL	ゼロックス	ART II, III
C A S I O	CAPPL/BASE		

- ・ 国際標準・業界動向

世界的には、Post Script、PCLといったものがデファクト・スタンダード。これらもカラー対応などに機能を拡張して、Post ScriptLevel 2、PCL L/5に進化している。

国内では、他社のプリンタとの差別化を図れなくなるためか、群雄割拠した状態。シリアルプリンタでは、PR201、ESC/Pページプリンタでは、LIPS、ESC/PAGE

が普及している。

(3)問題点・要望事項

① ハードウェア・インタフェース

- ・ 高速で利用者に優しい(プリンタの状態を簡単にきめ細かく制御できる)インタフェースが望まれる。
- ・ コネクタの形状(ピンの数)、結線が装置固有の接続状態になっている場合があり、標準化が望まれる。
- ・ 用紙カセット、トナー、リボンなど各社まちまち。共用が望まれる。

② ソフトウェア・インタフェース

- ・ 制御コマンドに一貫性がない。そのため、アプリケーションを複数のプリンタに対応させるべく、逆に、プリンタごとに備えている機能が異なるため、複数のプリンタに対応する場合には、個々のプリンタの機能を引き出しにくい。
- ・ 罫線コマンドなど、高速化に効くコマンドを使わず、イメージデータとして転送しなければならない。国内においては、罫線は必須機能。罫線込みの印字速度はユーザーに不満多い。
- ・ 外字・フォント・オーバーレイなどについては、各プリンタまちまち。これらはユーザー要件であるが、ユーザーごとに標準化(統一)し、個々のプリンタに定義/登録するのではなく、統一管理ができ、それぞれにダウンロードできる標準化されたミドルソフトの充実が望まれる。

③ ヒューマン・インタフェース

- ・ 設定/操作手順の統一
用紙サイズ、方向、縮小/拡大などの設定/操作手順/表現方法が各社まちまち
- ・ 障害時のリカバリ方法の提供と統一
ジャムリ発生の場合の再開などのページ単位での印刷機能の提供
- ・ 印字位置の統一
上下/左右のマージの取り方がプリンタごとに異なる。同じアプリケーションでも利用するプリンタにより印字バランスが崩れてしまう
- ・ 印字順序の変更機能の提供
優先度の高い印刷データが発生した場合、印刷順序を変更できる機能なども、業務処理では望まれる
- ・ カタログ表示の標準化
実行速度の性能表示、利用制御コード、ハードインタフェースなど

2.5 日本語入力

日本語入力の問題点としては、当部会で検討した、2.1「画面関連」、2.2「キーボード」等、HCI(Human Computer Interface)、操作性の問題もあるが、本項では、これらの問題については検討の対象外とし、かな漢字変換、ワープロのFEP(Front End Processor)としての利用に焦点をあてて報告を行う。

かな漢字変換にからむ問題としては、変換アルゴリズム以外に、装備している漢字・特殊記号の字数、文字の書体の種類等、いわゆる「漢字辞書」にからむ問題がある。「漢字辞書」については、その他、ソートの問題、地名や人名等の各種辞書の装備状況、ユーザー登録文字の取り扱い、辞書の機種間での交換性等も含まれる。

異機種間での辞書の交換性については、既に問題が提起されており、日本規格協会内のINSTACの「文書処理標準化委員会」では、テーマとして取り上げて作業をしており、最終的にはJISを作成することを目指している。ただし、現在はかなりしぼりこんだ範囲の検討を行っており、すべての辞書内の項目について、完全に異機種間での交換ができるというわけではない。辞書の完全な異機種間交換性の実現は、困難な課題であるが、必要性を感じる人はあり、徐々にではあるがその活動が行われていることを紹介するのみで本問題については、これ以上詳しくは言及しないこととする。

漢字辞書の種類、ソートの方法(一般に辞書内の文字にソートの情報がついているため辞書の問題とした。なお、ひらがな、カタカナのソートについては、上記のINSTACで検討を行った案がある)、ユーザー登録文字の取り扱いについても、本報告では詳しく取り扱っていない。

以上のように、本報告書では、かな漢字変換アルゴリズムを中心の話題とした。

(1) かな漢字変換アルゴリズムの状況

① 標準化作業の状況

現在知る範囲では、かな漢字変換アルゴリズムの標準化作業は行われていない。また今後標準化の対象になるとは考えがたい。

ユーザーとしても、かな漢字変換アルゴリズムの標準化によるメリットは少ないと考えられ、標準化を要望する必要はないと考えられる。

なお、言語間の「機械翻訳」の技術の中に、日本語文章の「クリーンアップ」、「文章の構文解析」という技術があり、この手法が、かな漢字変換に使用できる可能性がある。一部のワープロでは、この手法を取り入れていると考えられるものも存在するが、すべてのワープロにこの手法が取り入れられるには時間が必要であろう。

「機械翻訳」の手法についても、現在のところ標準化作業はない。

「機械翻訳」の研究の協同プロジェクトは存在するが、その結果がそのまま各種のワープロの、かな漢字変換アルゴリズムとして採用される可能性は低いと考えられる。

② 標準化の可能性とユーザーとしての考え方

かな漢字変換のアルゴリズムは、大きく次のように分類できよう。

- 1) 単語変換
- 2) 文節変換
- 3) 単文変換
- 4) 複文変換

ワープロの使用者からみた場合、単語変換のみでは不便であり、2)～4)の変換アルゴリズムが実装されていることが望ましいが、4)の複文変換(単文が複数個続いているという認識が可能で、前後の文との関連を考えた変換)は技術が未成熟であり、まだ標準化云々の話ではないと考えられる。

また、3)及び4)は前記のように機械翻訳の技術と密接な関係を持つと考えられ、機械翻訳技術が進展し、さらに機械翻訳技術とかな漢字変換の融合が図られることにより、飛躍的にかな漢字変換の質の向上が図られるものと考えられる。このような考え方が浸透することにより、標準化(というよりは、共通の手法を使用すると言う方が正確かもしれないが)の道は開けている可能性がないわけではないが、「機械翻訳」は、技術競争の対象となっており、標準化の可能性は(当面は)ないというのが実情であろう。

したがって、標準化の問題は、1)、2)となるが、1)は単語辞書の検索のみで達成されるため、アルゴリズム的には標準化とは無縁であろう。ただし検索の「学習機能」(同じかな読みがなされる多数の文字・単語があるが、使用者の利用に合わせて頻繁に使用される文字・単語順に検索していくよう検索アルゴリズムをシステム自身が記憶し変更していく機能)が必要であるという点では何らかの標準化ということは考えられるが、既にほとんどのワープロでは学習機能を装備しており、そのアルゴリズムが変わったからといってユーザーに影響はないのではないかと考えられる。したがって、この問題は各種の「辞書」の標準化、及び「辞書の異機種間交換」の問題に帰着すると考えられる。

③ 単文変換と学習機能の実験

変換の最初は、「名詞」+「接続詞(形容詞・副詞等)」+「動詞」のような形への文の解析である。もし、このように文節に解析することが可能であれば、「形容詞・副詞」や「動詞」の活用形の辞書の持ちかたが問題となり、やはり「辞書」の問題に帰着する。ただし、このような「文節」の解析技術も機械翻訳の範疇と考えられ、既に述べたように、技術競争の的であり、標準化は困難であると考えられる。

単文変換すら、かなり難しいことを例で示しておく。

あるテレビのクイズ番組では毎週、表記は同じでも意味の異なる文を状況によりどう解釈するかという問題が出されている。以下、例文を使い、ワープロでどう変換されるかの実験を行ってみる。

例文を、(良い例ではないが)「あさまではたらく」として、実際に変換してみる。「浅間で働く」。これが私の現在使用しているワープロ／文書フロッピー を使用した時の結果である。これが気に入らないのもう一回変換してみると「朝まで働く」となった。こちらを正しいとして変換を終了させる。この後で、同じ文を変換すると「朝まで働く」と変換された。これは、使用しているワープロが持つ「学習機能」の結果である。前者の変換結果は、私の辞書には「浅間」が 名詞として登録されており、変換時に文の解析は行わず、「ロングストマッチ」(辞書の検索時に登録されているものと入力を比較していくが、このとき先頭から最も長い文字数で一致するものを選んでくる手法)というアルゴリズムが適用されたためと考えられる。上記の例では「あさま」に対応する「浅間」がもっとも長い一致する語であったために前記のように変換されたと考えられる。確かに、この変換は両者とも、ある文脈においては正しい文章である。

文の解析を行うのではなく、直に「ロングストマッチ」で変換を行うと、とんでもない文章に変換される可能性は高い。

私の使っているワープロに文句をつけたようだが、もしかすると簡単な文節解析の手法が入っているのかもしれない。

再度、前記の文章を変換してみる。「あさま」の変換が、「浅間」となったので、学習効果が続いているかを確認することができる。「朝まで働く」と変換された。「学習機能」の結果がまだ持続している。

ここで、いったん本文作成の作業を中断してみる。そして再度、作業を開始し、変換を行うと「朝まで働く」と変換された。

中断して別のフロッピーを使用して別の作業を行い、再度変換を行ってみる。「朝まで働く」。まだ「学習機能」が残っている。フロッピーに学習機能の結果を覚えているのではなさそうである。

ワープロの電源を落とし再度立ち上げ、変換を行ってみる。「朝まで働く」。私のワープロでは、「学習機能」が電源の切断後にも残っている。

すなわち、「かな漢字変換」については、標準化よりもいかにそのワープロが「学習機能」を持つかが重要なのではないであろうか。

ちなみに、私の現在使用しているものはパソコン上のワープロソフトであり、パソコンは、12 MBのメモリ、十分なディスク容量をもっているディスクドライブなシステムである。

例を続ける。

今度は、フロッピーベースのもっと小さなワープロ専用機で実験してみる。

「あさまではたらく」を変換してみる。「あさまではたらく」。全然変換しない。ロングストマッチのアルゴリズムさえ実装していないようである。長い変換がくるとあきらめてしまうのかもしれない。

「あさまで」ではどうか。「朝まで」。これが変換の結果である。

「浅間」を登録して、変換を試みる。「浅間で」と変換された。さて、ここでいったん作業を

止めて、再び行ってみる。

「浅間で」。同じ結果である。

一回電源を落として行ってみる。「浅間で」。同じ結果である。どうも電源を落とすだけではなく、システム自身の再立ち上げを行ってみる必要があるようである。システムフロッピーを強引に読ませてみた。「浅間で」。同じ結果である。

では、別の文書フロッピーで行ってみよう。「浅間で」と出てきた。「朝まで」と出ると考えたがどうしてこうなるのであろうか。

私の考えでは、ワープロ専用機であるため、何らかのメカニズムがハード内にあり、電源を完全に落としても単語の比較順序の記憶を保持しているようである。

内蔵しているものも含めて、電池等をすべて取り払ってしまえば記憶がなくなると思うが、ここまでの実験はできなかった。

「浅間」が、もともと登録されていたのかどうかのチェックを忘れたので、「あさま」を「夜間」と変換するように登録してみる。「夜間で」と出た。上と同じように実験してみる。「夜間で」。「夜間で」。「夜間で」。これでもみな同じ結果が出ている。やはり、記憶を保持しているようだ。どのくらいの期間記憶を保持しているものなのかは、実験が困難であり不明である(多分、内蔵電池が使えなくなるまで、記憶を保持しているのだらう)。

また、同じフロッピーで、同種の機種であるが、別の機械にかけて行ってみた。

「朝まで」。「麻まで」。「厚狭まで」。「あさまで」。これが全部の変換結果であり、「浅間で」も「夜間で」も出てこない。フロッピーには、どうやら情報を記憶していないようである。または私自身が本機種を使いこなしていないだけで、フロッピーに情報を記憶させ、別の機械を使うときに、その情報を機械に読み取らせる機能があるようである。

④ 本項のまとめ

ここまでの結果を以下に要約しておく。また提言は最後にまとめることとする。

- ・ かな漢字変換アルゴリズムの標準化は困難
- ・ 標準化でHMIを統一するよりは、「学習機能」により、その人に合わせた環境に対応していくワープロが望ましい

なお、既に述べたように「辞書」の標準化の問題があるが、これは「辞書」そのものを標準化するというよりも、使用者の登録する文字を含めた「異機種間での辞書の交換」の問題としてとらえたい。本件は、既にワープロの業界として認識し、作業が始まっているので、この結果を待ちたい。

(2) ワープロのFEPとしての使用

文書処理を行うワープロと、プログラミングやアプリケーションへの入力データを作成するエディタとは一般には機能が異なっている。これは、文書を作成するためには、文書の清書のための各種情報が必要であり、作成された文書のファイル内の情報の持ち方が、エディ

タが作成する(旧来のカードイメージの)ファイル構造との間に相違が生じるためである。両者はその目的／機能が異なるため、仕方がないと考えるべきかもしれない。

このような現象は、同じベンダーのシステム内でも生じている問題である。

システムベンダーがトータルシステムとして作成した場合は、ワープロをFEPとして使用する方法を検討し、実現できる可能性はあるが、独立ソフトウェアハウスや、流通しているフリーソフトでは、このような配慮は困難であろう。

しかし、例えばMS-DOSのシンプルテキストだけのファイルに変換すれば、多くのワープロ間で文書の交換が(文字コードを標準の範囲で使用していれば、概ね)可能であるように、文書処理に必要な情報を取り除いた、シンプルフラットファイル(一般的には、固定長順編成ファイル)を作成する機能があれば、ワープロをFEPとして利用する問題の解決は可能であると考えられる。

本件も、最後に提言の形でまとめることとする。

(3)その他の事項

「ひらがな」「カタカナ」入力ではあまり問題がないが、「ローマ字」入力の場合に、以上に述べてきたことと異なる問題がある。

「ローマナイゼーション」(ローマ字でどのように文章を綴るか、表現するかという方法論)という問題である。これは、ワープロを使う人の世代の違いにより大きな問題となる。さらに、ワープロの開発者の違いにより問題が出てくる可能性もある。

例えば、「斉藤」はどうなるか。ある人はサインに Saito を使用している。しかしこれを変換してみると「さいと」となった。「Saitou」と入力すると「斉藤」と変換できた。

上の例は簡単な例である。かな漢字変換でも「さいと」では変換できず、当然「さいとう」と入力するから比較的理解しやすい。しかし「じょ」は「jyo」とするのか、「jo」だけでよいのか等、特に、小文字(ゃ、ゅ、ょ)や促音(ん、っ)を含む単語のローマ字の入力方法にはなやましい問題がある。

本問題については、学問的な分野で標準化の検討が行われている(残念ながら最新の状況は不明である)。ただし、この標準化は情報処理の標準化の世界の外側で行われている(文献を取り扱う分野で最も検討が進んでいる)。

本件について、ワープロ各種の実体がどうなっているかは未調査であるが、問題は生じていないのではないかと(あったとしても、慣れの問題であり大きな問題にはなっていないのかもしれないし、大きな問題になっているが私が知らないだけかもしれない)。

(4)まとめ

「漢字入力」の問題点とその解決についての、一つの考え方を示す。ただし、調査を完全に行っているわけではないので、提言には、既に実施済みのものも含まれている可能性がある。

かな漢字変換のアルゴリズムの標準化は範囲外でもよい(長期の課題であろう)。ただし、自己学習機能は、それがいかなるアルゴリズムでもよいが装備することとし、ない場合は仕様に明記するべきではないか。

また変換アルゴリズムによって、上手な使い方が考えられるのではないか。使用方法の手引のみではなく「上手な使い方の手引」的なものを作成することはできないか。

文字、単語辞書のポータビリティを実現するための方法の検討が必要であろう。

(システムに作り付けの)基本辞書についてのポータビリティの実現は不可能としても、少なくともユーザーが登録する「文字」「単語」の異機種間でのポータビリティについては、ユーザーの負荷の軽減が図られるべきである。キーボード、HMI等の標準化については、ワープロとしては範囲外でもいたしかたない。

また、キーと機能のバインディングについても範囲外となるのではないかと考える。

しかし、これらについても標準化の可能性があるかないかを検討し、最小限の標準化(変換キーや、キーフェースが同じ場合は機能を合わせる等)が可能かどうかの検討を行う必要があるのではないかと考える。

さらに、ワープロをFEPとして使用するために、日本語文書処理システムにおいて、その動作するシステムの標準のファイル(一般には非構造単純ファイル)を作成するオプションを追加するか、または日本語文書処理システムにより作られた文書を標準ファイル形式に変換するコマンドを追加することが必要ではないか。またローマ字入力のため、仮名とローマ字の対応、促音の入力方法等ローマナイゼーションの解説が必要である。

2.6 ネットワーク関連機能

(1)目的・対象

ユーザーが情報システムのオープン化に寄せる期待は極めて大きい。昨今の主流となりつつある分散処理やダウンサイジングで特に求められるのは、システム間の相互接続や相互運用である。ネットワーク機器に関してオープン化というときには、上記の他に、機器の

相互接続、すなわち製造業者が違っていても各種ネットワーク機器が相互流用できるということも含まれる。

本項では、ネットワーク機器関連のうちユーザーインターフェースとして見えている部分、特にユーザーよりのネットワーク機器に関する標準化の状況などを中心に、その問題点と対応方法を概観する。

(2) ネットワーク接続

ネットワーク接続に関しては近年、TCP/IP(Transmission Control Protocol / Internet Protocol)による接続が多くなっている。この場合、発生している問題は標準化そのものの問題というよりは、その運用及び前提としている環境に対する意識の相違が原因となって表面化する形となっている。具体的には、

- ・ TCP/IPアドレスの誤使用等
- ・ MACアドレスの誤使用等
- ・ LAN接続装置の使い分け

本項では、TCP/IPに関連した話題について触れる。

① TCP/IP (Transmission Control Protocol/Internet Protocol)

TCP/IPプロトコルそのものは、OSIレイヤのレベル3、4相当であり、厳密には標準機関で制定されたものではないが、コンピュータ間通信のいわばデファクト標準として、Internet(注1)の普及に伴い、その利用を急激に拡大している。今では、ほとんどのUNIXマシンで標準的に採用され、DOS、Windows、Macintosh 向けの製品化も進んでいる。またメインフレームについてもTCP/IPを実装しているものが増えている。

TCP/IPの内、トランスポートレイヤに位置するTCPは、2ノード間の仮想通信回線(Virtual Circuit)を確立し、通信に際しての伝送エラー、データの紛失、順序の入れ替わり、輻輳等を監視して適切な対処を講じることにより、通信の高信頼性を確保している。一方、IPプロトコルはネットワーク間の相互接続を目的に開発されたプロトコルであり、通信相手を表2-6-1に示す標準アドレスに基づいて識別している。

表2-6-1

表2-6-1 Internetのアドレス体系

レイヤ	名称	名称/アドレス
5、6、7	情報処理 アプリケーション	エンド・ユーザー識別詞 ポート番号
4	トランスポート	プロトコル名称
3	ネットワーク	IPアドレス
2	LLC MAC	LSAP番号 MACアドレス
1	物理媒体	なし

問題となるのはIPアドレスであり、本アドレスの設定誤りに起因するトラブル(ネットワークの異常輻輳、ネットワーク全体のダウン等)が従来かなり起こった。IPアドレスは図2-6-1に示すように、ネットワークアドレス、ホストアドレス、サブネットアドレスからなり、3つの体系(クラス)がある。

図2-6-1 IPのクラス別アドレス体系

クラスA		
0	Network(7)	Local Address(24)
クラスB		
10	Network(14)	Local Address(16)
クラスC		
110	Network(21)	Local Address(8)

クラス	最大収容ネットワーク数	最大収容ホスト数
A	126*	16,777,124
B	16,384	65,534
C	2,097,152	254

*No. 0及びNo.127は割当済み

図2-6-1

図のアドレス体系の内、ネットワークアドレスに関してはボランティア的活動で行われており、SRINIC(SRI International Network Information Center)で一元管理している。ローカルな接続のみの場合は、必ずしも正式なアドレスを必要とするわけではないが、世界規模のInternetや、それと間接的にTCP/IPで接続されているネットワークとつなぐときは、正式に取得した方が無難である。なお、日本ではJPNIC(1991年12月設立のJNICが1993年4月改称)が行っている。

IPアドレスについては、先に述べたようにInternetが全世界的拡がりとなってきたため、32ビットでは不足するという事態になっており、その拡張が検討され始めている。

(注1)Internetの概要

インターネットは米国防総省高等研究計画局(DARPA)が1969年に行った特定のベンダーに依存しない標準コンピュータネットワーク(ARPANET)の研究で開発されたネットワークである。現在、約130か国1千万人以上が加入し200あまりの代表的大規模ネットワークと繋がっている。従来、少数の大学や先端的な研究機関を結ぶ研究者ネットワークの性格が強かったが、最近は企業が積極的に利用し、世界で最も広く利用されているネットワークとなっており、今後も急速な伸びが予想されている。インターネットには、情報提供のコンピュータが多数存在しており、自由に情報を入手することができる。代表的な無料のサービスとしては、archie、wais、www、gopher & (e)mosaic等がある。また、Telnetを用いて、インターネットに接続されているコンピュータに接続し、各種商用データベースサービスの利用が可能である(この場合は当然有料、利用のためには別途利用者識別番号=IDの取得が必要)。

② MACアドレス

MACアドレス(48ビット)の管理は標準機関(IEEE)が実施している。現在48ビットの内、前24ビットはメーカー名、後24ビットは各機器ごとの製造番号を表し、ネットワーク一意のものになるように定められている。また最初の1ビットが“0”のとき個別の装置のアドレス向けであることを示し、最初の1ビットが“1”のときは、全ホスト向け(multicast)であることを示している。従来、メーカー等の理解不足によるアドレス付与誤りでブロードキャストが異常に増加するケースが多発したこともあるが、今はほとんど問題も生じなくなっている。

本アドレスは、一般ユーザーはあまり意識する必要はない。

③ LAN接続装置

LAN接続装置は、LAN間接続、公衆網接続、高速デジタル回線接続などLANのネットワークインテグレーションに際し、その接続点において双方のプロトコルを変換・管理する装置である。LAN接続装置はその機能等により以下のものがある。目的に合わせ機器の選定を行うことが必要。

- ・ リピータ……LANの接続距離を単に延長するための装置(レイヤ1)。
- ・ ブリッジ……MAC(Media Access Control)、LLC(Logical Link Control) からなるレイヤ2(データリンク層)での中継接続装置である。Filteringなどの機能を有する。

- ・ ルーター……レイヤ3(ネットワーク層)での相互接続を可能とする装置。ルーティング(経路選択)機能 や通信相手を限定するアクセスプロテクト機能等を有する。
- ・ ブルータ……ブリッジとルーターの機能を合わせ持つ接続装置。
- ・ ゲートウェイ……トランスポート以上のプロトコルを含めてプロトコル変換を行いLANと異なるネットワークとの相互接続を行う装置。

以下に、LAN接続装置の機能を示す。

レイヤ 4～7	→	ゲートウェイ
レイヤ 3	→	ルーター
		} → ブルータ
レイヤ 2	→	ブリッジ
レイヤ 1	→	リピータ

上記装置については、誤設定や経路情報変更の遅延などで情報の不一致等が発生し、下記のような問題を引き起こすので注意が必要である。

- ・ リピータ、ブリッジ……局所的な事故がネットワーク全体に伝染
- ・ ルーター、ブルータ……ブラックホール化し、情報が伝搬しなくなる

上記の問題を避けるためには、フェールセーフ機能を実装した機器を購入すると共に、ルーターについては専用のものを用い、ホストをルーターとして使用することは避けることが望ましい。IPアドレスの中にブロードキャストを行うための表記方法があるが、UNIX4. 2BSDとUNIX4. 3BSDでは仕様が異なっているのもその理由である(標準的仕様ではall'1'であるが、UNIX4. 2BSDでは all'0'の古い仕様に準拠しており、かつそれが結構広まっている)。

④ TCP/IP上のアプリケーション

TCP/IP上の代表的なアプリケーションは以下のとおりである。

- ・ Telnet……仮想端末機能が実現な、他のコンピュータに文字ベースでログインするためのコマンド及びそのプロトコルである。日本では漢字コードの変換を行うものが主流である。
- ・ FTP……ファイル転送プロトコルであり、ファイル転送の相手ノードへ接続を要求すると、ファイル転送用とは別にプログラム同士の会話用VCを確立し、ファイル転送用の情報の転送を行い、その後ファイル転送用仮想回線を確立して転送を行う。このFTPとは別に、簡略化されたファイル転送プロトコルである TFTPがあるが、あまり普及はしていないようである。

- SMTP……2ホスト・ノード間でメールを送信するプロトコルである(SimpleMail Transfer Protocol)。

- SNMP……ネットワーク管理用であり、障害管理、構成管理、性能管理、セキュリティ管理及び会計管理からなる。概念的にはIBMのネットビュー(Net View) やOSIのCMIPと類似している(Simple NetworkManagement Protocol)。

この他、ネットワークファイルシステムとしてサンマイクロシステムズのNFSやOSFのAFS等がある。

(3)ISDN接続

PC、WSをISDNへ接続するためには、大別して以下の2通りがある。

- PC、WSの既存インタフェースを活用しターミナルアダプタ(TA)を介して接続
- PC、WSに内蔵させたISDNボードを利用する方法

① TAによる接続

1) 低速の非同期インタフェースのタイプ

RS232CポートをもつTAでは安定性の点で若干不安定要素がある(もともと19.2kbps程度までをカバー。電話回線用の非同期モデムで64kbpsの転送を実現しているものもあるが、伝送エラーが発生しやすいのでケーブルの長さや容量に十分留意が必要)。

2) 高速の同期インタフェースのタイプ

DDX-Cのような回線交換網に接続されていた装置や専用線に接続して使用することを意図。V. 11インタフェースやV. 35インタフェースをもっている。

* 現在の問題点

- 互換性のない速度変換

PCやWSの非同期RS232Cポートへ接続して使用するTAを相互接続する場合、速度変換の方式が異なると通信できない(V. 110とV. 120の2種類がISDNで規定。米国AT&Tでは独自使用DMI model)。

V. 110は主に日本で利用、米国では、120が主流。

- 情報要素に対する問題

ISDNにおけるポイント・ツー・マルチポイント接続する際の端末属性指定のための情報（伝達能力、低位レイヤ整合性、高位レイヤ整合性、着サブアドレス 等の情報要素からなる）が、この情報に対する解釈なり処理が各装置で混乱している場合がある（例えば、ある装置ではノンチェックでも他の装置が厳格に チェックしている場合あり）。

② ISDNボードによる接続

TAでは、PCやWSで接続した場合、転送速度の制限や、TAとPCやWSとの間では複雑なインタフェースを利用することが困難なため、ISDN本来の特徴である豊富なオプションの制御ができないなどの問題がある。これらの問題を解決するものが内部バスに直接接続するISDNボード方式である。

これらの目的にファイル転送があるが、ソフトウェアとボードと一緒に提供されるのが通例であり、異なるベンダー間や機種間での普遍的な相互接続を保証するデータ転送プロトコルが存在しない。

〔参考〕異機種コンピュータ接続のための条件

- ・ 接続インタフェースが同じ
- ・ 通信プロトコルが同じ
- ・ データ表現形式／データコードが同じ
- ・ 共通ソフトウェアツールが双方の環境にある。
- ・ 動作環境の整合がとれている
(片方がオンライン状態になっていない等がない)

(4) 入出力機器

問題となるのは以下の点である。

① 本体～キーボード間のプロトコル、コネクタ等が不統一

このため、使い慣れたキーボードを別のマシンで使用できない(詳細は2.2参照)。

② RS-232Cの接続コネクタが不統一

RS-232Cの接続コネクタの種類には、DB25(SUN, PC98等日本では多数)、mini D in-8、DB-9(IBM-PC)、RJ-11、RJ-45、MMJ等があり(実質的にはDB25、DB-9が大多数)、さらに、これらは様々な接続方法がある。なお、RS-232CはCCITT の. 24/. 28、JISのJIS-X5101等と機能的な互換性がある。電気的特徴についてこの3者間で大差はないが、機械的仕様においては相違点があり注意が必要である(例:JISではコネクタの固定ネジをミリネジとしているが、他の2者ではインチを使用、JISではDCE側のコネクタはメスで

固定ネジもメスとしているがRS-232Cでは特に規定していない)。これらは通信機器が、近年急速に小さくなり、背面パネルに大きなコネクタが設置できなくなったことも一因となっている。

③ キャラクタベース端末制御

- ・ キャラクタ端末制御プロトコルが(VT-100やVT-200等)多数種あり、キャラクタ画面の表示を伴うAPのポータビリティがない。
- ・ Cursesの標準は、かなり以前に作られたため、現在はほとんどの端末でサポートしているカラー表示の機能がない。
- ・ Cursesがサポートできないワープロやペンコンピュータ等が増加。
- ・ UNIX以外の環境ではCursesはサポートされていない(IBM3270等)。

④ マルチメディアデータフォーマット

- ・ PC系、MAC系、UNIX系で音声や画像のデータフォーマットが統一されておらず、表示できない。
- ・ ~1.5Mbps対応のMPEG1は標準化済み。~15Mbps対応のMPEG2は標準化中。

⑤ モデムのピンケーブル

- ・ ネジが日米で異なる。

(5)その他

ネットワーク上のサービス(例:VAN)等については、今後調査し報告することとしたい。

2.7 LAN、WAN

通信、計算機技術の進展に伴い、各事業所でローカルに運用されるLAN (Local Area Network) と事業所間を結合して運用されるWAN (Wide Area Network)が発達してきた。

また、現在はLAN、WANがようやく普及時期を迎えつつあり、今後、音声、画像、データを統合したマルチメディアへの対応を控え、高速、大容量の通信インフラに対する需要も大きく膨らみつつある。

このような中で、情報システムを開発構築するユーザーはシステムの構築を短期間に低コストで実現するべく、標準化された製品を導入してシステムを構築しつつあるのが現状ではないかと思われる。

現在は、LAN、WANの標準化に関して多くの提案がなされている段階であり、これら標準化された製品の採用をするにあたって、いくつか考慮すべき課題があると考ええる。

よって、ここでは各LAN、WANの種類ごとの課題を列挙し、それを整理し、課題の明確化を図り、考えられる対応を記述することとする。

(1) LAN

① Ethernet LANについて

LANについて最も有名なのは、米DEC、米Intel、米Xeroxの3社が開発したEthernetである。

この3社の仕様は社名の頭文字をとってDIX規格として現在でも使用されている。

また、LANの標準化作業はIEEE(米国電気電子技術者協会)PROJECT802(802委員会)にて1980年から始められ、Ethernetでも利用されているCSMA/CD(IEEE803.2)等はこの委員会で作成されたものである。

このように使用されだしてから10年以上経過したLANであるが、デファクトスタンダードとしての生いたちを国際標準としたために、いくつかの不整合がEthernet(DIX V2.0)とISO8802-3(IEEE802.3)に生じている。

1) フレームフォーマットの違い

Ethernetは複数の上位プロトコルを識別するためにタイプフィールドを設けているのに対しISO8802-3は上位層にOSIのみを意識して作られているため、Ethernetのタイプフィールド部分をレンダスとして取り扱う。

2) 物理インタフェース

物理インタフェースであるピン数及びピンアサイン(シールドピン)の取り扱いが異なる。

3) SQE機能の取り扱いの相違

SQE機能(トランシーバーの衝突検出機能が正しく動作しているか否かをステーションが確認できるように、一定のタイミングで衝突信号をステーションに対して送出する機能)は、Ethernetはオプション、ISO8802-3は必須としている。

② FDDI(Fiber Distributed Data Interface) について

FDDIは伝送媒体に光ファイバーケーブルを利用し、米国規格協会(ANSI)が作成しているLAN規格である。

FDDIは国際的に標準化されているLANとしては最高速度であり、この高速性を生かして主にバックボーンLANとして使用されている。

ここで、FDDIの規格は4つの部分(物理層媒体、物理層プロトコル、媒体アクセス制御、ステーション管理)からなり、ステーション管理(SMT)を除く3部分の規格化が終了し、ISOで国際標準となっている。

しかしながらSMTの規格化は遅れており、現在製品として出荷されたものは規格案ANSI規格案レベル6. 2を基に国内ではINTAP(情報処理運用技術協会)が実装規約としたものをベースに作成されている。

また、INTEROP'91、INTAPINE'91、LANEXPO'92において相互接続性の試験が実施された。

ここで話題となったのはやはりSMTの問題であり、相互接続不良の8~9割の原因がSMTと言われている。

現在においてはSMT Rev7. 2が発行されるに至り、Rev6. 2と7. 2の共存や管理方法を含めた相互接続試験が課題としてある。

③ ツイストペア型FDDIについて

ツイストペア型FDDIは伝送路の価格を下げるためにANSIにて標準化が進められているツイストペアケーブルを採用したLANである。これは基本的に物理層媒体の規格が光ケーブルからツイストペアケーブルに変更されるのみで他の変更はない。

しかしながらケーブルについては現在2つの大きな流れがある。

1) CDDI(Copper Distributed Data Interface)

HP社、AT&T社、アンガンバス社を中心に製品化をしており、ANSI標準ベースである。

2) SDDI(FDDI over STP STP :Shielded Pair Cable)

IBM社、チップコム社、モトローラ社を中心に製品化している。この2つの流れについてはANSIは案で94年中に標準化の見込みである。

④ 100MEthernetについて

100MEthernetについては現在2つの勢力が技術を競っており、同じ種類のものがそれぞれ規格化される可能性がある。

1) IEEE802. 12 : 100VG-AnyLAN

1992年11月にHP社、AT&T社が共同で提案した高速Ethernetで、音声級ケーブルの使用を前提とした技術である。

2) IEEE802.3 : 100BASE-T

1993年3月に3COM、インテル社SUN社等が提案したもので10MEthernetと同一アクセス方式でビットレートを10倍化した技術である。

両案については1995年前半にも標準化作業が終了する予定となっている。

⑤ 無線LANについて

無線LANの出現背景については、

- ・ オフィスへのLAN敷設工事ができない

OA用に設計されていない古いタイプのビル
賃貸ビル

- ・ 組織改編等による頻繁なオフィスレイアウトの変更
- ・ セミナー、展示会等の一時的なLANの構築

等への回答として、近年無線LANが欧米において実用化・製品化されてきた。

日本においては電波法の制約があり、ほとんど実用化されていなかったが、1992年12月の改正(2.4~2.5GHz帯の免許不要)により製品が発表されるようになった。

製品化動向としては、NCR(AT&T)とモトローラがシェアを拡張している状況(NCR:30%、モトローラ:18%)であり、代表2社の(NCR、モトローラ)の製品概要は以下のとおりである。

■モトローラ社(AltairJ)

*基本構成

- ・ CM(Control Moduler)と複数台のUM(User Moduler)で構成、UM1台当たり最大8台の 端末接続が可能
- ・ 1台のCMで最大50台の端末を管理可能
- ・ UM、CM間の伝送距離は最大40メートル

*既存有線LANとの共存

- ・ 実行速度5.7Mbps(伝送速度は15Mbps)とEthernetと同等の速度

- *ダイバシティ・アイテナの採用

- ・ CM、UM間で、36通りの伝送路を持っており、常にマトリクスの中から最良の伝送路を選択して使用

- *ネットワーク・マネジメント対応

- ・ SNMPエージェントをサポートしており、MIB-及び拡張MIBに対応

- *ブリッジ機能

- ・ UM内に接続された端末間の通信はUMのブリッジ機能により、そのUM内で終結するため、基幹LAN内への影響を最小限に抑える

- *セキュリティ

- ・ スクランブルとアクセス制限の併用

- NCR社(Mave LAN)

- *基本構成

- ・ 無線装置が組み込まれたWaveLANインタフェースカードと呼ばれるNI(Network Interface Card)と全方向型アンテナから構成
 - ・ 有線LANへの接続はNIC搭載サーバーまたはWavePoint機器を使用
 - ・ NICはAT互換機(ISAバス)及びMCA(MicroChanel)用を提供。この他NECがOEMし、PC-98やEWS4800シリーズで提供
 - ・ 有線LANとの接続は10BASE-5、トークンリングをサポート
 - ・ 接続距離はセミオープンで30~60メートル、クローズドで15~30メートル

- *ISMバンドの使用

- ・ 2.4GHz帯を使用し、耐干渉性の強いスペクトラム拡散を採用
 - ・ 伝送速度2Mbps、実効データ転送速度154Kbpsを実現
 - ・ 無線局のライセンスが不要

- *無指向性アンテナの使用

- ・ 対等分散方式に向いており、クライアント間のPEERtoPEER通信が可能
 - ・ ケーブルが不要であり、移動性に優れている

* 現存プロトコルとの互換性

- ・ 業界標準のネットワークドライバ (IPX、DLPI、NDIS、ODI) をサポートしており、Netware、LanManager、TCP/IPとの互換がある。

* セキュリティ

- ・ 盗聴、妨害、混信に強いスペクトラム拡散を採用
- ・ 独自の機密セグメントIDコードによる相手判別
- ・ 暗号化AES (Advanced Encryption Scheme) の採用

■ 標準化動向

国際標準の動向としてはIEEE802委員会で標準化作業中 ('94 末までには標準化予定) であり、国内標準については電波システム開発センターで検討中である。

ここで、現状では無線LANについて標準化されたものはなく、各メーカー独自に製品化しているが、標準化団体 (IEEE) で制定後は、規約に合致した製品が出回ると思われる。

また注意しておかねばならないことは、いままでに成立しているLANの規約はすべて有線であり、現在策定作業中のLANは無線であることから、セキュリティ管理方式が規約に取り込まれているかどうかという点である。

また、規約に合致した製品同士の接続についての保証 (規約には冗長性をもたせることがある) は標準化団体 (IEEE) で行われなため、ユーザーが独自に調査・実証を行わなければならない可能性がある。無線LANについてはIEEE802. 11 委員会にて標準化作業中であるが、現在はNCR (AT&T) とモトローラがシェアを拡張している状況にある。

⑥ TCP/IPについて

これまでの標準化は物理層とデータリンク層の規格であり、ネットワーク層以上のLAN固有のプロトコルは原則として存在しない。しかし、LANボードやLAN用アプリケーションが標準的に採用しているプロトコルがある。その代表的なものとして米国国防省 (DOD) のTCP/IP (Transmission Control Protocol / Internet Protocol) がある。

TCP/IPが広く利用されている理由は仕様が公開されLAN用プロトコルとして採用できる環境が整っていたことと、UNIXが標準でサポートしたこと等が考えられる。特にアプリケーションレベルのプロトコルであるFTP (ファイル転送) や SMTP (電子メール) なども標準化されており、簡単な業務なら特別な通信ソフトウェアを利用しなくとも実行できる。

ICMP (Internet Control Message Protocol) は IP モジュールの一部として実装されているプロトコルである。機能としては IP フレームの異常に対して、その異常を送信元に通知する機能と TCP/IP 動作上必要な情報入手のための問い合わせメッセージであるが、ICMP を受け取った際の各端末の動作について記述されていないため製品によっては受信ステーションに無視されるケースがある。

⑦ SNMP (Simple Network Management Protocol) について

SNMP は TCP/IP ネットワークの標準管理プロトコルとなっている。もともと OSI 管理の CMIS/CMIP (Common Management Information Service /Protocol) の標準化が進むまでの短期解として開発された。しかしながら TCP/IP ネットワークが広く普及したことや、OSI 管理がなかなか標準化されなかったことなどから急速に利用が広がっている。

ここで SNMP を利用するにあたり管理情報収集のため多数のパケットのやり取りが発生し (ポーリングベースであるため) ネットワークのトラフィックを増やしてしまうことがある。

また、MIB (Management Information base) : 管理対象の集合が標準 MIB (IBA (Internet Architecture Board) が規定している) と各ベンダーが独自に定義した拡張 MIB の 2 つあり、各メーカーにおいてすべてサポートされているとは限らない。

(2) WAN

① フレームリレーについて

フレームリレーについては、近年国際的には ITU (International Telecommunication Union) における ITU-T 勧告があり、国際的には TTC 勧告が進められている。TTC 勧告は基本的には ITU-T をベースとした内容となっており、ITU-T 準拠で推進されている。国内においては、1994 年 6 月より NTT においてサービスインが予定されている。

このような中においても、ITU-T と ANSI での相異が見られる (DLCI (データリンク識別子) の使い方等) ことや、ITU-T では物理インタフェースは I. 430 か I. 431 を規定しているが、現実には製品化されている交換機は X. 21 や RS422 等であるので事前にチェックが必要であったりする。その他においてもフレームリレーサービスでは網が輻輳した場合の回避方法については明確にされていなかったり、課金方法についても明確でない。

② ATM (Asynchronous Transfer Model) について

ATM の標準化については、ITU-T や民間団体の ATM フォーラムなどで進められている。ITU-T は B-ISDN の実現を目的として ATM を定義しているのに対し、ATM フォーラムは LAN や自営 ATM 網への実装も対象としている。ATM フォーラムは ITU-T の勧告があればそれをベースとして取り込み、未勧告でも早期に必要な内容については ITU-T と連

携をとりながら仕様を作成している。よってITU-Tより仕様作成作業が進んでいる部分もかなりある。

このような中において、ITU-T勧告はいくつか出されているが、内容についてはあいまいな部分が多く、まだこれからとの感が強い。

(3) 課題の整理

以上のような課題を整理してみると、

- ① 規格(規約)の中にあいまいさを含んでいる部分がある。
- ② 規格(規約)は決まっているが、ユーザーがオプションパラメータを選択可能で、結果としてうまくつながらない。
- ③ レイヤの低い部分は当然標準化していかなければならないが、ユーザーが利用する第7層(アプリケーションレイヤ)の充実がTCP/IPの例を見ても必要である。
- ④ 標準規格の設計は技術を有するベンダー中心のためユーザーニーズを反映することが難しい(メーカー間の戦略に振り回される結果となる)。
- ⑤ 世界的コンセンサスを得るため標準作成に時間がかかる。
- ⑥ WANの分野についてはITU-T勧告、LANの分野についてはOSI、IEEE等が作成しているため、これをシームレスに接続また管理するための方策を考える必要がある。

などが考えられる。

(4) 改善策ならびに考え方

規格(規約)の曖昧さについては、ハードウェアについては導入時における仕様への反映や、検査で顕在化 させることができるが、ソフトウェアについては導入時検査で発見する方法しかとれない。テストを実施するにあたっては時間、費用もかかることから、その効率的実施方法を検討しておかなければならない。幸い最近になってOSI one(メーカーの任意団体)がインターオペラビリティテストの結果のDB化を推進する動きが見られることや、OSIのコンフォーマンステストに Internet も加える動きがあることから、そのような情報の利用やFD DI製品のテストについてはANTC (Advanced Networking Test Center :米国)を利用するなど検討し、テスト工数を削減すべきと考える。

また導入するソフト製品の仕様を作成する上においては、どの規準を満足すべきか明示しておくことも必要と考える。

以上のように、情報の共有化を図り、情報システムに関わる人たちの生産性向上を図ることが大切である。

また、無線LANの中でも周波数の割当等の法規制の問題があったが、今後さらなる規制の緩和が必要と思われる。

WANについては今後、音声等のリアルタイム系とデータ系の融合が図られてくると予想できるが、現在は、シームレスな融合のためのシナリオを早く描く必要があると考えられる。

最後に、新しい製品や規格(規約)については前項の、のとおり、ユーザーはアプリケーションサービスの享受が最終目的であるから、ユーザーは特にこの部分に対して積極的に意見を述べることである。これによって標準への反映も現在よりも進展していくと考えられる。

各分野について思うがままに課題を書き出し、その課題のすべてについて改善策は示すことができなかったが、システム設計／構築していく上で参考となれば幸いである。

2.8 ドキュメント・データ標準

ドキュメント・データについては、身近なワープロ等のOAソフトで作成するケースが最も多いが、スタンドアローンOA機用が開発されてきた経緯からわかるように、このような領域では、機能重視、優先のため個々のソフト間のデータ形式を統一するのは、当分の間、非常に困難なことと思われる。

本項では、既に統一されたデータ標準を目指し活動が続けられている技術分野の動向を参考に、ドキュメント・データ標準に関する今後の課題を検討した。

(1) 現状と問題点

① 現状

OA機器の大幅な普及、適用業務の拡大と深化は、電子的に処理される情報、特にドキュメントの割合を急激に増加させている。対象となるデータ形態は文字(テキスト)以外にイメージ、図形が追加され、テキストフォントの増加も合わせ、多様な表現様式を持つに至っている。

例えば、契約書、見積書、予実算管理書、ビジネス参照資料、取り引き関係書類、コミュニケーション資料、技術文書、プロジェクト共通文書などでは、通常の事務手続き文書以上に多様で高品質な表現を要求され、ドキュメントのデータ構造もより複雑になっている。

その作成手段は、内部データ構造の異なるパソコン、EWS、大型ホスト、専用ワープロであり、利用されるソフトもワープロ、表計算、DB管理ソフト等多くのものがある。さらに技術分野ではCAD、CGのようにビジュアルなデータ形態もドキュメントの一つとなる。

このようなデータがネットワーク利用の進展に伴い個人レベルで使用するワープロや表計算ソフトから 全社的な事務、技術関連のアプリケーション、あるいは企業間で相互に関連するようになり、伝達、参照、利用される機会が多くなるに従い、互換性のないデータ形式が情報の共有、流通の大きな問題として顕在化してきている。

② 問題点

一過性のデータであればある程度我慢できるが、企業内のドキュメントはそれ自体がデータベースとしての役割の側面があり、伝達や再利用の際の電子化されたデータの互換性は重要な問題である。

具体的に問題点を列挙すれば以下のようになる。

- 1) ソフト間でデータ形式が異なる。特に国内のパソコン用ワープロではデータ変換の対象はテキストに限定されている。
- 2) 機種間で電子化媒体が異なる。例えばフロッピーの初期フォーマット形式が異なる。
- 3) データ形式が異なるため、プリンタ等の出力媒体で文字フォント、サイズ、罫線等の線種が一様にならない。
- 4) イメージや図形が統一して扱えない。
- 5) 日本語入力が統一されていない。
- 6) 機種、ソフトにより日本語のコード体系が異なる(JIS、シフト JIS、EUC)。
- 7) ソフトごとにドキュメント作成操作が異なる。
- 8) ドキュメント内容は作成されたソフトを起動しないと確認できない。
- 9) ドキュメントの変更管理が容易でない。
- 10) 異なるソフト間でE-mailが利用できない。
- 11) データ圧縮技術が活用されていないため、ドキュメントデータ量が膨大になる。

以降では、主にドキュメントの形態がより複雑となる技術分野を対象に、企業全体の商品が形成されるサイクル全体を通じたドキュメントのかかわり合いから検討を進める。

(2) 必要要件

このようにソフトごとに作成されるドキュメント形態が不統一な状況に対して、ユーザーが業務でドキュメントを作成する場合に必要な要件は、電子情報化社会において際限なく湧き出て、一朝一夕では解決が困難と思われるが、連動するドキュメント作成ソフトや周囲の処理環境と共にこうあってほしいと願う姿をあげれば、以下のようなになる。

① ドキュメント内容、作成機能

- 1) 文字フォント、サイズが容易に定義、変更でき、ソフト間で相互に参照できる。
- 2) 多国語に対応している。
- 3) イメージ、図形等を含む。
- 4) 複数ドキュメントの編集(切り貼り)が容易にできる。

② 表示・出力

- 5) どのような環境(機種・フォント)であっても一様な表示、出力が可能である。

③ 変更、管理

- 6) ドキュメント作成変更が最小限の作業で行える。保管、検索が容易で履歴が保存できる。
- 7) ドキュメントの変更内容が記録され、かつ同時に共同利用が可能。
- 8) 一般アプリケーション、OAソフト、DBMSから出力されたドキュメントファイルが容易に互換でき、
他のソフトのファイルが参照、引用できる。
- 9) 大容量ドキュメントの保管・取り出しが迅速に行える。

④ ネットワークでの利用

- 10) 作成されたドキュメントが社内及び企業間で標準化され、ネットワーク上で利用可能。ネットワーク通信、配信に対応している。
- 11) ネットワークからの参照に対して機密が保持される。

(3) 標準化の動向

国内では、ソニーが提唱し推進している文書互換のための標準フォーマットCDFPに20数社のベンダーが参加している。これはUNIXワークステーション(ソニーNEWS)の文書を対象とし、テキスト、ベクトルデータ、イメージデータ(ラスター)を含む文書構造(レイアウト、文書構成、データと属性)を規定している。一般OA用のワープロは対象外である。

技術分野でのドキュメントデータの標準化については、国際的な活動が活発になりつつあり、企業内での生成と企業間でのデータ交換の2つの観点から整備が進んでいる。

その1つはCAL(S (Computer-aided Acquisition and Logistic Support)計画で、米国安全保障産業委員会と国防産業が共同で推進している。CAL(Sは、設計開発段階から、製造、保守、運用、教育訓練に至る各段階で発生する情報の標準化を図り、データベース化により情報の再利用をもくろむもので、トータルなライフサイクル・コストの削減、さらには設計等の時間短縮を計ろうとしている。

戦闘機的设计製造には多くの企業が係わり、関連する技術ドキュメントは数10万～100万ページに及ぶ膨大なものとなり、製品の変更改良やそれに関連する技術伝達、教育で多大な労力が発生すると言われている。そのため85年に米国防省で正式な計画として改善の検討が開始された後、標準化の作業に移り、第1段階の89年～94年に情報を1つのデジタルファイル交換標準に統一し、94年以降の第2段階で情報のデータベース化を行う予定となっている。

CALS計画の実現によりエンジニアリング領域で期待される効果には、設計変更時に80%の時間削減、図面アクセスが数日から数時間へ、図面枚数が200枚から3枚へ、教育訓練時間が82%削減、実物模型が不要等があげられており、遠望な計画が進みつつある。

CALSでは表2-8-1のデータ標準を採用／規定している。

表2-8-1
(ファイルが見つからない)

ドキュメントに直接関連するSGMLは、1986年にISO 8879として承認され、92年にはDOD(米国防省)への納入の際に義務づけられる等、標準として定着している。SGML文書は、文字コード体系など全体を定義するSGML宣言、論理構造を定義する文書型定義(DTD)、実体を表す文書インスタンスで構成され、タグで個々の内容を識別する。出力や表示にはスタイル定義(DSSSL、FOSI)、プリンタ言語(PDL)が組み合わされる。単にドキュメント作成の目的だけではなく、文書データベースにも応用されている。

CALSは米国の国防における情報取扱規約を出発点として、ISOのような対応する国際標準を取り込み標準が策定されているが、すべてにわたってビジネスで流布している国際的な包括規定に対応しているわけではない。今後、技術ドキュメント標準として実際に効力を発揮するためには、現実的な標準の取り込み、移行が必要と考えられている。例えばPDES/STEPのように標準化が初期の段階にある規定を採用しており、安定するまでにかかなりの期間を要するものと予想される。

米国以外でも、日本、ニュージーランド、オーストラリア、韓国、台湾、シンガポール、西欧において広がりを見せ始めているが、まだ評価段階にある。

表2-8-2 CALS 関連規約

表2-8-2

表2-8-2 CALS関連規約

領域	軍用規格	包括的基準	内容
機能仕様	MIL-STD-1388-1 MIL-STD-1379D MIL-STD-2167A		LSA:後方支援分析 軍の訓練ソフトウェア
データ	MIL-STD-1388-2	PDES/STEP (ISO 10303) EDIFACT/ANSI X.12 (ISO 7535他)	製造データ交換 LSAR:後方支援分析記録 ED17フォーマット
共有/ ソフトウェア環境	MIL-STD-1388-2 B		IRDS : 国際R&D,標準化 GOS/P : 政府ソフトウェアシステム概要 SQL : DB照会言語 RDA : 研究、開発、調達 POSIX : OSインターフェイス互換性
データ交換	MIL-STD-1840 MIL-D-28000A MIL-M-28001 MIL-R-28002 MIL-D-28003 その他	X.400 ISO 8879他 ISO 8613 ANSI X3.122	技術情報交換 IGES:CAD図面交換 SGML:文書記述言語 FCII, DSSSL, SPDL CCITT G3, G4, Titled G4 :ラスターデータ (FAX, CAD) CGM:ベクターデータ HPGL, ASCII, TIFF等

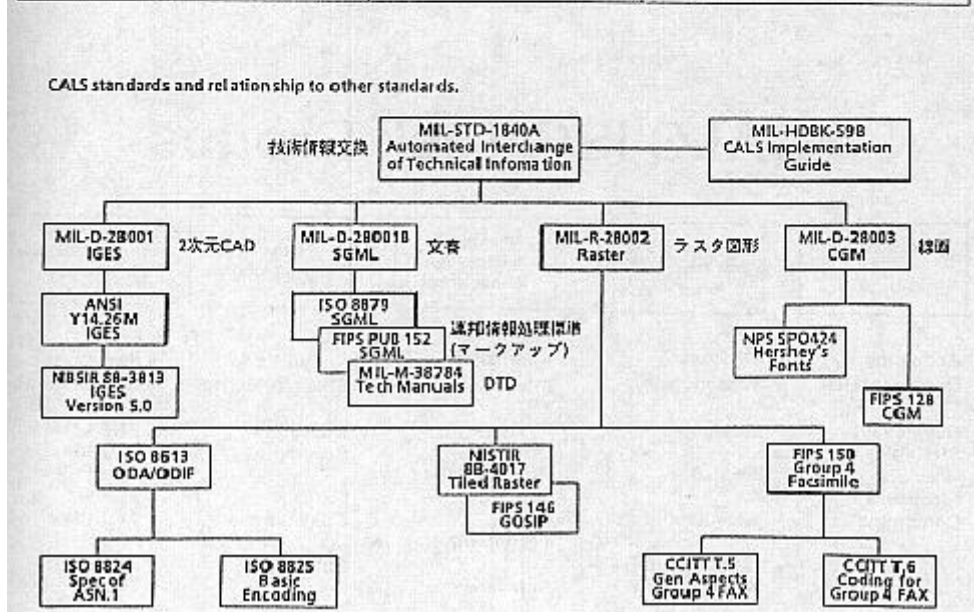


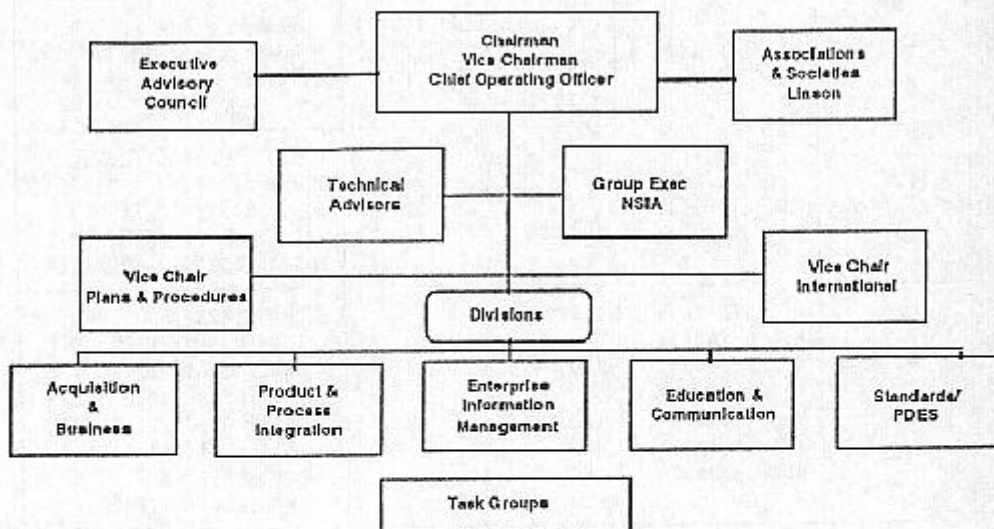
表2-8-3 米国での CALS 推進組織

表2-8-3

表2-8-3 米国でのCAL S推進組織

ISG : the CALS/CE Industry Steering Group (CE:Concurrent Engineering)

CALS ISG Structure



CALS ISG Task Groups

Acquisition & Business	Product & Process Integration	Enterprise Information Management	Education & Communication	Standards/PDES
Acquisition Documentation	Process Management	Information Integration	Suppliers & Small Business	Standards Administration
Business Strategies	Design	Systems Design & Production	Education Programs	PDES Activities
Electronic Commerce	Logistics	New Technology	EXPO & Conferences	EDI Activities
	Manufacturing	Software	Public Relations	
	Configuration Management			

(4) 要望

ドキュメント・データは企業の中核のノウハウを含み非常に重要な情報源になるべきものであるにもかかわらず、その標準化は軽視されている。データの標準化を全面に押し出しているOAソフトは一部を除き一般には出廻っていない。ドキュメント・データを生成するソフトに関連するベンダー各社にはもっと前向きな姿勢で対応していくことを切に望みたい。

このような観点を踏まえ、いくつかの要望事項を以下に示す。

- ① ドキュメント・データの標準化への参画を意志表示する。
- ② 中間フォーマットとしてのデータ標準の協議機関を明確にする。
- ③ 国際標準化機関への積極的な参画(ISO等)。特に日本語コードの国際規約への対応として国内の意見を集約し提示する。
- ④ 個々のソフトについては準拠しようとしているデータ標準と範囲を明確にする。

当面、行ってほしいこととしては、

- ⑤個々のソフト間のデータ変換機能を提供する。
- ⑥そのためデータ形式、論理構造を公開する。
- ⑦日本語コードの統一(JIS、シフトJIS、EUC)。

困難であればコード変換ツールの提供。

一方、ここ最近、音声や動画、映像を含むいわゆるマルチメディアの利用が急速に進んでいる。今後、情報 伝達内容は、従来のドキュメントのように静的な文字主体のものから、さらに多くの情報を含むイメージを多用したものやマルチメディアを対象としたものへ移 ることが予想される。

そこでは多くの情報をもつため、大容量データの高速転送や保管が大きな課題となり、データ圧縮技術が必須の技術となる。したがって、JPEG(カラー静止画像圧縮)、MPEG(カラー動画圧縮)、MHEG(映像、音声を含むマルチメディアデータ圧縮)の ような国際標準にも留意したい。

第3章 相互接続性・相互運用性・可搬性関連

3.1 TPモニタ

(1)TPモニタ関連の標準

TPモニタ関連の業界標準としては、以下のようなものがある。

- ・ X/OPEN DTPモデル
- ・ OSF DCE
- ・ ISO OSI TP標準
- ・ IEEE POSIX 1003.11 TP標準 等

ここでは、TPモニタの内部のコンポーネント及びコンポーネント間のインタフェースを規定しているX/OPEN DTPモデルについて考察を加える(図3-1-1)。

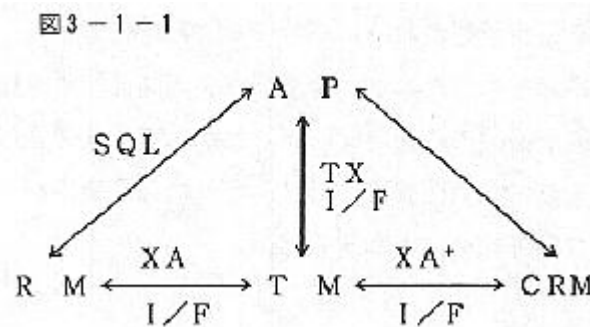


図3-1-1

RM(RESOURCE MANAGER)は、共有資源(DBMS、プリンタシステム等)に対するアクセスを提供する。この部分はTPモニタには含まれない。

TM(TRANSACTION MANAGER)とは、トランザクション処理の開始や終了等の処理を行い、TPモニタの主要部分である。

CRM(COMMUNICATION RESOURCE MANAGER)は、クライアント/サーバー間のコミュニケーション機能を提供し、TPモニタの一部である。

このモデルは、汎用機のTPモニタと比較すると、TPモニタ特有の機能がデータベース機能から分離されている点が大きく異なる。そのため、既存のデータベースを比較的自由に組み合わせることが可能になっている。しかし、その反面、障害時等に、TPモニタのログと、DBMSのログと両方みる必要がある等、運用が複雑になっている部分もある。

TPモニタは、1つのサーバーで大量のオンライントランザクション処理を行う場合には必須の機能であるが、通常のクライアント／サーバー型のアプリケーションを構築する場合も非常に有効なツールである。

(2) X/OPEN DTPモデルの標準化動向

APとRM間のインタフェースは、RMがRDBの場合は、SQLインタフェースが事実上の標準である。

APとTM間には、TX I/Fというプレリミナリー仕様がある。

TM-RM間には、XAインタフェースがあり、これは正式な標準になっている。

TMとCRM間には、XA+インタフェースという標準の検討が行われている。

APとCRM間のAPIは、現在、標準の検討が行われていて、TUXEDOベースのXATMI、EncinaベースのTxRPC等が標準の候補となっている。

したがって、このモデル自体、未完成の状況であり、アプリケーションの可搬性等が保証できるレベルには至っていない。早期に確定することが望まれる。

(3) 標準化の問題点

分散アプリケーション開発には、TPモニタは様々な意味で重要である。

- 1) OLTP処理の実現
- 2) 通信制御処理部分の開発工数削減
- 3) アプリケーション開発の生産性向上
- 4) システム運用負荷の軽減、等

X/OPEN DTPモデルは、上記の1)2)やAPの可搬性を高めるという意味では効果があるが、異種のTPモニタを相互接続するということは現状の標準化のレベルでは困難である。異機種のTPモニタ間の接続には、OSI TPがCRM間の標準として有力である。

また、アプリケーション開発環境については、一部COBOLや4GLがTPモニタ配下で使用可能なものもあるが、メインフレーム環境と比較すると、必ずしも高度な性能が提供されているわけではない。

システム運用に関しては、TPモニタによりターゲットとしている環境に差があり、機能レベルは製品によりまちまちであるが、システムの分散化という流れを考えると、より高度な運用機能を実現することが望まれる。

今後の標準化において、TPモニタが備えるべき運用や開発のための標準機能の定義とそのAPI等の標準化が一番必要と思われる。

〔例〕・運用支援機能

クライアント管理／構成変更／ソフトウェア配布／定義体作成支援

・開発支援機能

テスト支援／COBOL／デバッカ

(4) 製品の標準化準拠状況

X/OPEN DTPモデルが、XAインタフェースのみ正式標準という状況なので、各TPモニタは独自仕様のAPIを定義している。

TUXEDO、ENCINA、TOP END等の主要なTPモニタの製品群は、X/OPEN DTPモデルへの準拠ということの方針として打ち出し、積極的に標準インタフェースをサポートする姿勢を示している。

RMとして、XAインタフェースをサポートしているDBMSは増加している。

ORACLE、SYBASE、INFORMIX等が最新版で、XAインタフェースのサポートを行っている。TPモニタによっては、XAインタフェースをサポートしていないDBMSが利用できるように、特別なインタフェースを提供しているものもある(TOP ENDのXAベニア等)。

3.2 データベース

昨今のワークステーションやLANの発達とクライアント／サーバーシステムの進歩により、分散システム環境がますます利用されるようになってきている。このような環境において、異機種分散環境下のデータベースを容易に利用できることが、ユーザーの要求するところである。そのためにはデータベースに関連する様々な技術の標準化が必要である。本項では、データベースの標準化についてこれまでの経緯と今後の動向を紹介し、ユーザーから見たデータベースについての問題点を提起する。なお、本文中、* で示した番号は、本項の終わりに掲載した参考文献を指し示すものである。

(1) ユーザーのデータベース使用状況 (*1)

X/Openが行ったXtra調査(注)によると現在最も多く使用されているデータベースはSQL対応のリレーショナルデータベース(RDB)であり、将来依存度が高まると考えているデータベースはオブジェクト指向データベース(OODB)である(表3-2-1、2参照)。

	全体 (A)	アメリカ	ヨーロッパ	日本	現在使っていないユーザーの割合 (B)
SQL	60%	91%	89%	60%	17%
オブジェクト指向	13%	23%	11%	6%	80%
SQL以外のリレーショナル	51%	62%	44%	47%	44%
階層型	69%	79%	67%	61%	30%

表3-2-1

	現在使用中のユーザー (Aの内数)	現在使っていないユーザー (Bの内数)	全体 (左記の合計)
SQL	71%	10%	81%
オブジェクト指向	9%	56%	65%
SQL以外のリレーショナル	10%	3%	13%
階層型	7%	0%	7%

表3-2-2

(注) 1993年にX/Openが、北アメリカ、ヨーロッパ、日本、オーストラリアにおいて実施した、237名のオープンシステムのオピニオン・リーダーを対象に行った調査。なお表中の「アメリカ」には、アメリカ、カナダ、オーストラリアが含まれる。

以下、各データベースの標準化の概要と、現状及び将来主流となると考えられるRDB及びOODB関連の標準化動向について記す。

(2) データベースの標準化動向

① これまでの各データベースの標準化概要 (*2)

1) 階層型データベース／ネットワーク型データベース

データベースに関する最初の標準化は、1967年米国データ・システムズ言語協議会CODASYL(The Conference on Data Systems Languages)によって行われたネットワーク型DBMSについてのものであり、階層型DBMSについては行われなかった。階層型DBMSの主流であるIMSをリリースしているIBMは、CODASYLが階層型ではなくネットワーク型を使用したことに強く反発、「IBMはあくまで階層型で行く」と宣言しており、現在でもCODASYL準拠製品をリリースしていない。

2) RDB (*3)

RDBの標準化は、その操作言語であるSQLの標準化を中心に行われている。これまでのSQLの標準化経緯を以下に記す。

海外	
1986	ANSI(米国規格協会)で米国標準化
1987	ISOで国際標準として制定「SQL(IS9075-1987)」→SQL1
1989	ISOで SQL1 機能拡張を制定 「SQL1Addendum(補遺)1(IS9075-1989)」→SQL89 追加機能 ① 参照制約定義機能 ② 定義域を制約する CHECK (検査制約)と DEFAULT (既存値の指定)機能
1992	ISO規定(大規模DBシステム構築対応) 「SQL(IS9075-1992)」→SQL2→SQL92 追加機能 ① データ・タイプの追加 ② 表やビューの削除/変更が可能 ③ 動的SQL
国内	
1987	JIS規格制定「JIS SQL(X3005-1987) 」
1990	JIS SQL 補遺(X3005-1990)
1993	SQL92 対応版 JIS SQL(X3005-199X) 原案作成済み 現在規格調査委員会での審議待。

1) その他のデータベース

その他データベースの標準化は行われていない。

オープンシステムの標準化を推進している X/Open では、SQL92 と Informix Corporation により発行された C-ISAM 製品のためのインタフェースを採用している。

② 主要DBの今後の標準化動向

1) RDB (*3)

SQLの今後の標準化について以下に記す。

- SQL92 について提案に上がっている追加規定
 - a. コール・レベル・インタフェース(CLI) →SQL92 補遺1の公算大
 - b. ストアド・プロシジャ機能→SQL92 補遺2の公算大
 - c. トリガ機能→SQL9X (以下 SQL3)または SQL92 補遺3の可能性あり
- SQL3 の計画

ISO データベース言語(DBL) グループ会議でSQL標準化プロジェクトの見直し提案
 →このままでは規定が大きくなりすぎて実現が困難(1992.11)
 「SQL3 では、規定を機能ごとにパッケージ化して個別に標準化作業を進めていく」

- 提案事項

*SQL3 では、規定をコアとパッケージ部分に分ける。
 *コア部には、SQL89 、及び「SQL92 の中間規定(intermediate+ α)」を採用する。

+ α 部分は、整合性機能と各国語対応を付け加える。

*パッケージ部は次の5機能で構成する。

- ① ランゲージ・バインディング
- ② 演算処理機能
- ③ トランザクション機能
- ④ データ・タイプ
- ⑤ セキュリティ

- DBLグループ会議でオブジェクト指向とマルチメディア対応の機能を SQL3 に取り込むことを決定

- オブジェクト指向に関する追加機能

- ①クラス
- ②クラス階層と継承
- ③メソッドの記述
- ④抽象データ型のサポート
- ⑤複合オブジェクト
- ⑥オブジェクト識別子

- ・マルチメディア対応

SQL3 そのものの機能ではなく、SQL3 から参照するライブラリのような位置づけとして考えている。

「SQL Multimedia and Application Package(SQL/MM)」

→SQL3 対応のマルチメディア機能の標準化プロジェクト

1993 年秋に、SQL/MMをISO の標準化プロジェクトとして取り上げるよう日本を中心に提案がなされた段階であり、まだ今後のスケジュールなどは決まっていない。

- ・問題点

* DBL グループはSQL3とSQL92 の非互換は避けたいとしているが、SQL3をSQL92 の延長上からはずしたため、すべての項目に互換を確保するのは難しい。

* 計画では、SQL3 を1994 年中にDP(draft proposal)化し、1995 年にDIS(draft international standard) として確定することを目標にしているが、検討項目の多様さを考えるとその時期の達成は微妙である。

2) OODB (*4)

OODBに保存するオブジェクトは、永続性(Persistence) をもたせるための指定が必要であるが、この指定は、永続性をもつオブジェクトを継承する方法や、ポインタを使う方法があり、統一されていない。OODBに関する標準推進団体ODMG(Object Database Management Group)が標準仕様を作成中である。

SQLの標準化に関連して、ANSI SQLの標準化でオブジェクト指向の要素を取り入れる方向にある。これは1995 年に規格化されるはずのSQL3で 実現する。SQL3で検討されている項目には、オブジェクト識別子、抽象データ型、データ型の拡張、継承による階層化などがある。

OODBMSと呼べる製品はRDBMSを包含していると見てとれる。しかし、オブジェクト指向言語が一般に広く受け入れられてない点や自由度が高い点で、扱いの難しさが指摘される。一方、RDBMSは複雑なデータ構造を扱う面で機能不足が指摘されている。RDBMSのオブジェクト指向拡張とOODBMSの SQL採用によって、お互いの欠点を克服しつつ接近してきている。

[参考]標準化関連団体概要説明

① ISO(International Organization for Standardization) (*5)

工業及び技術に関する国際規格の統一と国家間の調整を目的として、1947年に設立された国際機関で、90か国以上から国家を代表する標準化団体が加盟している。日本からは日本工業標準調査会が参加している。

② CODASYL(The Conference on Data Systems Languages) (*6)

1959年、アメリカの政府、コンピュータ・メーカー、ユーザーの3団体によって構成された、事務用プログラミング言語の開発を目的とする団体。1960年には事務用プログラミング言語としてコボル(COBOL)が発表され、その後、データベースの言語についても研究が行われ、コダシル型データベースを開発している。

③ ANSI(American National Standards Institute) (*5)

アメリカを代表する民間の標準化団体。アメリカを代表してISOに加盟しており強い影響力を及ぼしている。アメリカ政府機関もANSIに加入している。

④ X/Open (*5)

1984年、フランスのブル、イタリアのオリベッティ、イギリスのICL、旧西ドイツのシーメンス、ニクスドルフといった欧州の中心的企業5社により、UNIXの統合化を目的として結成された。現在はUI及びOSF自身とその主要メンバーのすべてを含む日・米・欧のベンダー24社、及びユーザー、ソフトハウスなど、約120社の企業、団体から構成されており、特にヨーロッパにおいて強い基盤をもっている。

(3) 異機種データベース分散環境のための標準化動向

異機種データベースを、分散環境で自由に利用するために標準化団体及び業界団体で様々な活動が行われている。

ここでは、ISOが標準化に取り組んでいるRDA(remote database access)、及び業界が推進しているODBC(Open Database Connectivity)、IDAPI(Integrated Database Application Programming In terface)について記す。これらは、その対象範囲、実現環境は違うが、図3-2-1のRDAの例に示すように、クライアント側とサーバー側(もしくはアプリケーション側とDBMS側)のAPIを規定するものである。

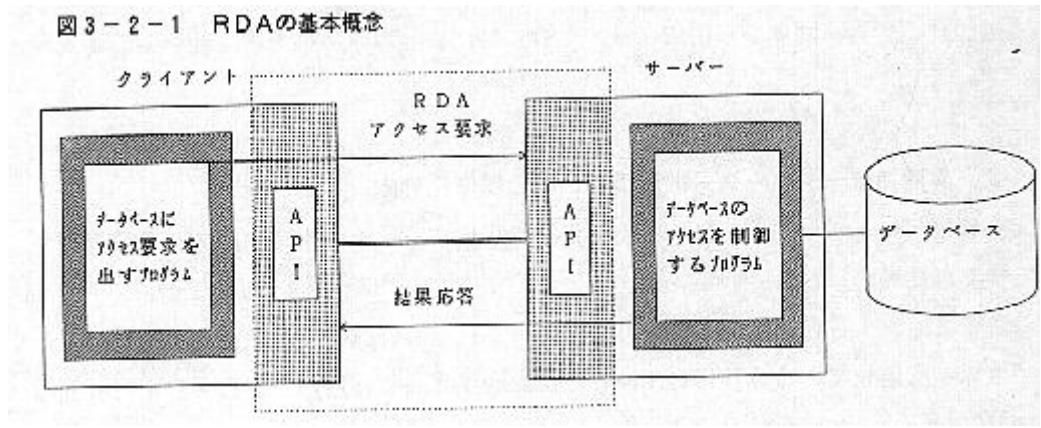


図3-2-1

① RDA(remote database access)の標準化動向 (*7)

複数ベンダーのデータベースにまたがる分散データベース・システムを構築するために必要な、共通したデータベース及び通信インタフェースを提供するのがRDAである。以下にその標準化動向を記す。

海外	
1985.11	OSI参照モデル第7層のアプリケーションの一つとしてISOで標準化開始
1985.12	ECMA(欧州電算機製造者協会:European Computer Manufacturers Association)のデータベース技術委員会が技術報告書を提出。
1987.12	規格草案作成
1992.06	ISO、RDAの基本標準を国際規格(IS)化
国内	
1988.11	INTAP(情報処理相互運用技術協会:通産省管轄の社団法人)主催のOSI相互接続実験デモが世界で初めて実施された。
1993.03	INTAP、RDAのIS化を受けて、RDA実装規約の初版を完成。

今後の動向

ISOは、SQL92 への対応を含めた標準 RDA の機能拡張を検討しており、1995 年6月を目指して改訂版(Amendment) のIS化を予定している。

② ODBC(Open Database Connectivity)の動向 (*8)

クライアント側のアプリケーションからデータベースにアクセスするための標準APIとしてマイクロソフト社が提唱している仕様。91 年 11 月米国で発表。

ORACLEなどの主要RDBベンダーはODBC対応を表明しており、現在各社ともODBCドライバを開発している。

<ODBC対応の Windows アプリケーション>

「ACCESS」(マイクロソフト)

「Excel」(マイクロソフト)

「VisualBasic」(マイクロソフト)

「PowerBuilder」(ニチメンデータシステム)

*RDAとの違い

RDAはOSIを前提としているが、ODBCにはその限定はない。

RDAは機能を定義しており、APIまでは規定していない。ODBCは Windows 環境のAPIである。

③ IDAPI(Integrated Database Application Programming Interface) の動向(*8)

ボーランド、ノベル、IBMなどが提唱している標準インタフェース仕様。ODBCを包含する位置づけでありSQLベースのAPIと非SQLベースのAPI とを分けたアーキテクチャとなっている。現在仕様を詰めている段階であり、具体的なプロダクトの登場は 94 年夏頃と見られている。

④ ユーザーから見たデータベースの標準化についての問題点

以上、データベースの標準化について記してきたが、最後にユーザーから見たデータベースの標準化についての問題を4点提起する。

1) 各DBMSベンダーは、他社との差別化のために標準に加えて付加機能を付けるが、どこまでが標準の

範囲でどこからが付加された機能であるかを明示していないため、非標準仕様を使ってしまうことが多い。

2) APIが標準化されると各DBMSの差異はコストパフォーマンスのみとなるが、コストパフォーマンスを比較

する客観的データが不足している。DBMSについて業界標準の性能評価基準として、TPC (Transaction

Processing Performance Council)の定めた各種のテストが存在するが、この基準は測定環境までを規定し

ておらず、測定はメーカーに委ねられている。そのため現状TPCベンチマークテスト結果として公表されて

いるものは、メーカー自身が自社製品に有利な環境で測定した結果が多く、実システムに直接使用するこ

とはできない。

3) 企業の基幹システムでは、高グラフィック、高レスポンス、高信頼性の要求があったり、RDBでは扱えない

ような長大レコードが必要になる場合が多い。そのような場合には階層型DBMSで対処しなければならな

いが、これに対する標準化の動きがない。

4) ODBCやIDAPIのように、国際標準ができる前に複数の業界標準ができてしまうため、ユーザーは標準

化前の選択を迫られてしまう。国際標準化をもっと迅速にすることが必要である。

[参考文献]

- *1 オープンシステムへの指針 X/Open 1993 ([本文に戻る](#))
- *2 データベース管理システムの現状(日経データプロ・ソフト 1989.8 NS3-500-201) ([本文に戻る](#))
- *3 SQLの標準化動向(日経データプロ・ソフト 1994.2 NS3-500-251) ([本文に戻る](#))
- *4 オブジェクト指向DBMS 試用期から導入期へ、ツール類の整備も進む
(日経オープンシステム 1993.10 p.135) ([本文に戻る](#))
- *5 オープンシステム入門(末松千尋著 ダイアモンド社) ([本文に戻る](#))
- *6 わかりやすい コンピュータ用語辞典(高橋三雄監修 ナツメ社) ([本文に戻る](#))
- *7 RDAの標準化動向(日経データプロ・ソフト 1993.10 NS3-500-501) ([本文に戻る](#))
- *8 ODBC環境 API標準化で変わるミドルウェアの役割
(ネットワークコンピューティング 1994.4 p.56) ([本文に戻る](#))

3.3 オペレーティングシステム(OS)

(1) OS発展の経緯と特徴

1940年代後半にコンピュータが世の中に現れてから、数多くのOSが世の中に送り出されてきた(表3-3-1)。OSの役割をまとめると、「コンピュータシステムを構成する物理資源(ハ

ードウェア資源)、論理資源(ソフトウェア資源)を管理し、資源操作上の複雑な事項(資源ごとの多様な特性やその利用法)を隠蔽し、ユーザーに使い勝手のよさを提供すること]になる。

表3-3-1

表3-3-1 OS概念の発展

年代	システム	概念	主要な流れ			
1950	IBM701用JOシステム IBM704用GM/NAAシステム	最初のOS 人出力サブルーチン ジョブの逐次処理				
1955	SHARE OS FORTRAN モニタシステム SAGE SABRE	アセンブラの提供 高水準言語の提供 リアルタイム処理 トランザクション処理				
1960	DOS ATLAS OS/MFT, OS/MVT CTSS, DTSS, TOPS-10 MULTICS TIE	人出力制御システム (IOCS) ジョブ制御言語 逐次バッチ処理 割込み システムコール機構 仮想記憶 スプーリング タイムシェアリング (TSS) セグメント方式仮想記憶 ダイナミックリンク 階層的ファイルシステム リダイレクション リンク保護 高水準言語によるOS記述 階層構造OS セマフォ	バッチ処理	リアルタイム機能	TSS	
1970	TSS/S RSX/15 CP/CMS, VM/370 UNIX	ミニコン用タイムシェアリング ミニコン用リアルタイムOS 仮想マシン シェル (コマンドインタフェース) フィルタとパイプ 移植性				
1975	CP/M MVS VMS ゼロックス ALTO, STAR	最初のマイコン用OS IBM370用仮想記憶 VAX用マルチユーザサポート 専門家向けワークステーション アイコンとマウス				パーソナルOS
1980	MS-DOS マッキントッシュ PICK VRTX, MTOS SI	16ビットマイコン用OS グラフィックワークステーション 関係データベースOS 軽装リアルタイムOS ビルディングブロック型OS		超リアルタイム機能		
1985	SUN NFS OS/2 TRON	ネットワークでのファイル共有 マイコン用マルチタスクOS 標準リアルタイムOS体系 遠用分野対応OSの系列化 作法の統一 多国語サポート 流通性 仕様の公明				標準化
1990	windows NT COSE	ネットワーク機能 マルチメディア、GUI				

OSの最もプリミティブな機能は1950年代初頭にバッチ処理をより効果的に行うために考案され、汎用大型コンピュータで発展してきた。その後、ミニコン、PC、WSの出現に合わせてその機能が引き継がれると共に、より効果的に処理を行ったり操作するための新しい概念が考案・提供されており、特にPC分野では使いやすさを追求したOSが登場している。

コンピュータはあらゆる分野に導入されており、OSに要求される機能は極めて多岐にわたり、そうした要求を実現するためのOS機能も要求のレベルにより異なる。このため、標準化の動きも適用分野により異なった動きにある。以下、次に挙げる項目について、標準化の動向・問題点等を述べる。

- ・ 汎用情報処理分野

バッチ処理、オンライントランザクション処理などの伝統的なビジネス用情報処理システムとして利用。

メインフレーム、WS等に搭載されているOSが対象。

- ・ パーソナル処理分野

多数の人間が使うことを前提としないセルフユースとして利用。PC、携帯端末等に搭載されているOSが対象。

- ・ リアルタイム処理分野

原子炉監視・制御などのように、瞬時で、かつ高信頼の処理を必要とする分野。交換機や工場の監視・制御装置から、通信システムなどの装置に搭載されているOSが対象。

なお、上記の分類のうち標準化が進展しているのは、汎用情報処理のWS分野に限定されているのが実状であり、それ以外は各種コンソーシアムやユーザー団体等により提案普及しているデファクト(事実上の)標準の位置づけになっている。

(2) 汎用情報処理分野の標準化の動向

OSの標準化が一番進展している分野である。この分野は歴史が長いということのほかに、ユーザーが多くオープン化に対する要求が一番大きいことも、標準化活動を促進したと言える。特にWS分野では、UNIX(注1)の展開に合わせて標準化が図られてきた。一方、メインフレームでは長らくIBMの製品が支配的であったが、IBMコンパチ路線を採用するメーカー以外は独自のOSを持ち、IBMを含めOSをユーザーの囲い込みの道具とするために非公開としたことにより、OSそのものの共通化はさほど進展していない。このため、特定の目的を持った企業が集合し(コンソーシアムを構成し)OSインタフェース(注2)なり製品に関する共通化の検討開発を実施しているのが実情である。なお、この分野での標準化機関以外の代表的な団体・機関は以下のとおりである。

- ・ UNIX International (UI)……実質上消滅
- ・ Open Software Foundation (OSF)
- ・ Common Open Software Environment (COSE)
- ・ X/Open

- Network Management Forum (NMF)
- Petroleum Open Software Corporation (POSC)
- Group of 10
- Advanced Computing Environment (ACE)

(注1)UNIXはX/OPENカンパニーリミテッドがライセンスする、米国及びその他の国における登録商標。

(注2)OSという言葉は、ソフトウェア製品を示す場合に用いられる。これに対して、OSの機能や使い方を規定した部分使用のことをOSインタフェースと呼ぶ。

① UNIX仕様の共通化

AT&Tにより開発されたUNIXは、その使いやすさと共に当初無料でリリースされたことなどから急激な拡がりをみせた。しかし、ユーザーが独自に手を入れられることから互換性のないUNIXが氾濫した。その内の大きな流れとしては、AT&TのSystemVと、カルフォルニア大学バークレー校のBSD(Berkley Standard Distribution)の2つがある。このような状況を解決するため、1984年米国IEEE学会にUNIX標準化を目的としてPOSIX(Portable Operating System for Computer Environment)委員会が設置され、その検討結果がISOに提案され標準化が行われた(現在も進行中)。1985年にはX/OPENが結成され、応用プログラム(AP)の流通と相互接続を図るための仕様を検討し始めた。

業界側でも1987年にAT&Tとサンマイクロシステムズが提携し、SystemVとBSD統合のUNIXの開発に着手し(後のUI)、これに反発したIBM、DECなど7社が1988年に対応組織(OSF)を結成し、互いに主導権争いを演じた。この争いは、昨年夏、米国ノベル社がAT&TからUNIXの開発元であるUSL(UNIX System Laboratories)を買収したことにより終焉し、次いでUNIXの商標がノベル社からX/OPENに委譲された。今後はX/OPENが規定するUNIXの共通OSインタフェース仕様に基づいた製品のみがUNIXと呼べることになる。この意味ではユーザーにとって極めて好ましい方向にある。

なお、UIとOSFを構成する主要メンバーがUNIXシステム上でソフトウェア実行環境を共通化するCOSEを結成した。最近1~2年のこうした動きは、米国マイクロソフト社のWindows NT(注3)への対応策との読みが一般的である。

最後に、ユーザーへの注意を一言。UNIX(というより、パッケージソフト全般)は系列の違いによって仕様が異なるのは通例であるが、同じ社についてもバージョンが変わると互換性を保証しないのが一般的である。システムの移行に際しては十分留意が必要である。最近では、Solaris(注4)での混乱が記憶に新しい。

(注3)Windows NTは、米国マイクロソフト社の登録商標。

(注4) Solaris はサンマイクロシステムズ社の登録商標。

② ユーザー主導によるOS共通化

上記はベンダーが主導となって、いわば土台である下の方から進めている共通化の動きである。これとは別にユーザー自身の要求である応用プログラム(AP)の流通性、相互接続性、相互運用性、保守の簡素化等の観点からOSインタフェースを共通化しようとする、いわば上の方からの動きがある。

前述したように、メインフレーム系のOSインタフェース仕様は、各社とも独自仕様に基づいた製品を出荷してきたことにより、オープン化の時代にあつて、APの相互流通性、システム間の相互互換性、さらにはデータベース(DB)をはじめとしたシステム間相互運用性の点で問題があつた。具体的には、新しいシステムの導入に際しての膨大な量の(特にトランザクション系の)AP・DBの修正移行作業、システム間相互接続に際しての通信プロトコル整合検証・試験作業、システムの運用操作方法や運用データの相違によるオペレータ要員等の再教育等である。

このような機関・団体のうち、有力と思われるものに、NMFの SPIRIT (Service Provider Integrated Requirements for Information Technology)がある。これは Bellcore、BT 及びNTTなど世界の主要なキャリア(通信電話会社)と主要コンピュータベンダーが共同で、汎用コンピューティングプラットフォームのためのキャリア共通仕様を開発しようというものであり、昨年3月発足、9月に第1版を発行した。1995年3月末には最終仕様発行の予定である。これに先立ってNTTと主要ベンダーとの間で開発されていたMIA (Multivender Integration Architecture)については、そのほとんどの仕様がそのままSPIRITに採用される方向である。

POSCは、1990年に世界の5つのメジャーが中心となって設立した、石油探索と生産(Exploration and Production)のためのコンピュータ技術を集約し、P&Eのためのオープンなソフトウェアプラットフォーム仕様開発と製品開発を目的としたコンソーシアムである。現在5メジャー以外に40社余りのハードウェアベンダー等が参加している。

GROUP10は、1991年に米国の10大企業により結成された、ベンダーにオープンシステムの提供を迫るためのコンソーシアムであり、オープンシステムアーキテクチャへ誘導するための標準仕様とプロトコルの検討開発を目的としている。

(3) パーソナル処理分野

PC等に代表される処理分野に適用されるOSの特徴は、基本的にはシングルユーザー、シングルタスクの機能と言える。最近では、マルチタスク機能に拡張される傾向にある。本分野における標準化機関はなく、またコンソーシアム等も従来あまり多くなかつた。MAP/TO

P (Manufacturing Automation Protocol / TechnicalOffice Protocol) やEISA (Extended Industry Standard Architecture)はその例である。しかし、最近では、マルチメディアに対する期待の増大・技術の発展に併せて、各種情報端末や携帯端末などの分野で多くのコンソーシアムなどが形成されている。

この分野のいわばデファクト標準となっている(なっていた)ものは以下のとおりである。

- CP/M (Control Program for Microprocessors) (注5)

マイコン上で動作する最初のフロッピーディスクベースのシングルユーザー、シングルタスクのOS。

高水準言語PL/Mで記述され、ハードウェア依存部の基本入出力を分離したことにより、各種のプロセッサに搭載され、8ビットマイコンのデファクト標準となった。

- MS-DOS (MicroSoft Disk Operating System) (注6)

16ビットパソコンのデファクト標準となっているOSである。メモリ保護機能がない、メモリ容量が640K バイト、等の制約がある。

- OS/2 (Operating System/2) (注7)

MS-DOSの制約を解消すると共に、マルチタスク機能、グラフィック機能/ウィンドウ機能などを持つOS。

なお、この分野では、シングルユーザー、マルチタスク機能を持つと共に、高速の応答性を実現するためのリアルタイム機能を加味しビットマップディスプレイをもつWSやPCのためのOSインタフェースとしてBTRON (Business TRON)があり、トロン協会で仕様が検討されている。

(注5) CP/Mは、米国デジタルリサーチ社の登録商標。

(注6) MS-DOSは、米国マイクロソフト社の登録商標。

(注7) OS/2は、米国IBM社の登録商標

(4)リアルタイム処理分野

電子交換機、産業用ロボット、発電所等の監視・制御システムなどにおいては、緊急の処理を即座に実行する必要があり、厳しいリアルタイム性が要求される。このため、従来はAPとOSを一体化したシステム作りや専用のリアルタイムOSの利用が中心である。しかし、標準化なりオープンなものがまったくないわけではない。比較的汎用性の高いものとして、VRTX (Versatile Real-Time Executive) (注8)とMTOS (Multi-Tasking Operating System) (注9)

等がある。しかし、両者とも緊急時の処理速度や提供機能がプリミティブすぎることによるAP開発量、及び信頼性等の点の問題点が指摘されている。

上記の問題を解決すると共に、ソフトウェアの流通性を狙いとしたリアルタイムOSインタフェース仕様がCTRON(Communication and Central TRON)であり、トロン協会で検討が行われている。先に述べたPOSIXでもリアルタイム拡張仕様が出ているが、上記のような分野に適用する上ではまだ機能不足との指摘がある。

(注8)VRTXは、米国レディシステムズ社の登録商標

(注9)MTOSは、米国インダストリアルプログラミング社の登録商標

3.4 分散処理

『オープンシステム』『ダウンサイジング』といった大きな流れがある。

ここで、オープンシステムの“オープン”とは製品の設計仕様に関する情報がオープンであり、このために標準化が促進され、ユーザーにとってベンダーや機種を組み合わせ等の選択の幅が広がることになる。したがって、オープン化、ダウンサイジングの特徴は仕様統一のための標準化活動が活発に行われ、オープンな分散処理を実現する環境が整ってきたことである。ここではユーザーの立場で「分散処理環境」に関する標準化の動向と、分散処理環境を進めるにあたって、その課題について述べる。

(1)分散処理環境

① 分散処理環境の概要

「分散処理環境」とは計算機の使用形態を指す用語であり、「マルチユーザー型」「ネットワークコンピュータ型」、あるいは「クライアント/サーバー型」と呼ばれている。システムにおける計算機などの資源を、ネットワークを活用してより効率的に利用できる環境を目指すものである。具体的には「計算機等の各装置にそれぞれ特定の機能をもたせた機能分散化や、効率的な資源配置を行って負荷分散化を図ることにより、高いコスト・パフォーマンスを追求すること」、そして「それら各装置をネットワークを通して共有し、高い効率を求めること」が上げられる。これを実行するためには強力な通信機能を備えていることが前提である。

② 情報システムの発展段階(図3-4-1参照)

図3-4-1 集中型から分散型・統合型へのシステム形態の発展

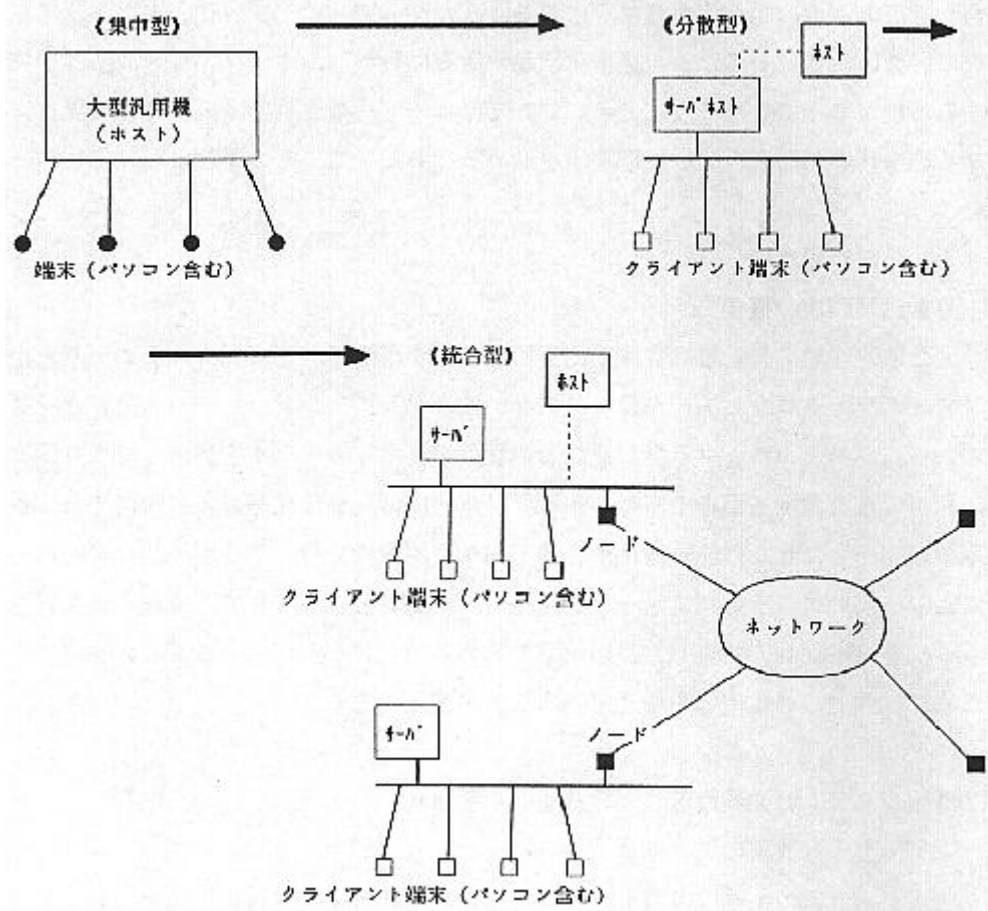


図3-4-1

1) 集中型→分散型のメリット

- ・ システムの柔軟性の向上:各ユーザーのニーズに合致したシステム設計が可能
- ・ 操作の生産性の向上:汎用的であるための不必要な処理操作を排除する
- ・ システムの発展性:段階的かつ部分的発展を実現
- ・ 負荷分散による効率的資源活用:応答時間に対しての最適資源の配分
- ・ システムの信頼性の向上:リスク分散によるシステム稼働率向上
- ・ 運用時間の制約が少ない

2) 分散型→統合型のメリット

- ・ 資源共有による効率的資源活用:データ、電子メールなどの単純な情報資源交換から、共有性の強い資源(ソフト)、特殊性の強い資源(共有プリンタ)の活用
- ・ 機能分散によるコスト・パフォーマンスの向上:各構成装置が特定機能に特化することにより高いコストパフォーマンスを達成。

- ・ 負荷配分による効率的資源活用:常に全体として最も効率の高い状況に負荷を自動的に配分する
- ・ 水平分散機能の充実:分散処理のための通信ソフトの充実。
仮想ワークステーション、リモートリソースアクセス、ファイル転送、アプリケーション間通信、遠隔システムの監視、障害回復を行う「遠隔保守」などワークステーション同士の水平分散機能の充実。

③ 統合型分散処理の考え方

- 1) 独立して運用・管理されている複数のコンピュータシステムをネットワークで接続し、プログラム、データ、周辺装置などの資源を相互に利用できるようにするもの。
- 2) 複数のコンピュータシステムをネットワークで接続し、全体で単一のシステムであるかのように見せるもの。

(2)分散処理環境に関する標準化作業

マルチベンダー環境でメインフレームやUNIXシステム、パソコンなどを統合して分散処理システムを構築することが可能になった。ネットワークで統合されたコンピュータの複合体はユーザーからは単一のシステムに見える。データの所在地や処理がどこで実行されるかなどユーザーは意識する必要がなくなってくる。

オープンな分散処理システムを構築するために必要な環境の標準化作業が着実に進展しており、基盤となる製品の出荷も始まった。

- ・ OSF (オープン・ソフトウェア・ファンデーション) : UNIXの標準化推進団体が分散処理環境の基本機能をパッケージ化したDCE (分散コンピューティング環境)を提供している。メンバー企業のIBMや日立製作所などはUNIX環境だけでなく、メインフレームなどの独自アーキテクチャにもDCEを搭載している。
- ・ UI (UNIXインターナショナル) : UNIXの標準化でOSFと対立関係にあったが、94年3月、UIの主要メンバーである、AT&T、サン、富士通、ノベルがOSFに大同団結、今後コンピュータの標準化作業が一本化されることになった。

分散処理環境のフレームワークを規定したATLASを提供してきたが基本的にはDCEをサポートしている。サンマイクロなどが設立したCOSE (共通オープンソフトウェア環境) やPST (既存技術) プロセスと呼ぶ標準化作業基盤を設立した。CDE (共通デスクトップ環境) や共通AP (アプリケーション・プログラミング) 等PSTで新たなソースコードの開発ライセンス提供等、すべての作業がOSFに集約されることになり、オープンシステム志向を強めるユーザー側の要請に向けた標準化作業が進展すると思われる。

・ X/OPEN: メーカーから中立なオープンシステムへの将来の方向付けを行うことを目的にオープンシステム標準仕様XPG4を発行している。X/OPEN がUNIXの商標権を取得、従来UNIXの標準API仕様は一本化されていない。そのためベンダー間でAPの互換性が保証されていなかった。

今後はX/OPENのブランド認証制度がスタートすると、SPEC1170（共通API）への準拠が保証される。

オープンな分散処理環境を実現する上での基準となる仕様は、以下のとおりである。

- ・ユーザー側の要求事項に立脚しているか
- ・移植性と相互運用性の両方をカバーしている
- ・特定のメーカーから独立している
- ・特定の製品から独立している。
- ・国際的か
- ・継続的に改訂／拡張が可能か
- ・実用的か(多くのメーカーの多数の製品に適用されているか)
- ・中立的な検証手段があるか

(3)分散処理環境の期待効果

① 期待効果

様々なオペレーティング・システムや処理環境のプラットフォームがネットワークを介して混在できる環境が求められるようになった。このような環境で構成されるコンピュータ間で資源を共用し、協調した分散環境でアプリケーションを効果的に動作させる以下の機能を提供する必要がある。

1) リソースの共用(分散コンピューティング;接続可能性)

分散コンピューティング環境でのリソース供給者は供給者とユーザーの位置に無関係に、一様で透過な方法によってスタンドアロン環境の場合よりもより広範囲なユーザーのもとへ情報を届ける。分散コンピューティング環境は認可されたカスタマーであれば何の制限もなくリソースにアクセスしたり、逆にリソースを提供したりすることが許されている。単一の分散コンピューティング環境で唯一仮想的に実現できる利点は並列化である。リソースは同時に、しかも並列にアクセスできる。

2) 先端技術の利用

近年のマイクロエレクトロニクス分野における最新の先端技術を使うことができる。したがって、分散コンピューティング環境を構築することはコスト的に効率のよい方法である。またシステムの段階的な投資も可能である。

3) 可用性と信頼性

重い負荷がかかったり部分的な障害が発生しても、システムサービスとリソースは利用可能で、保護完全性は維持される。サービスが無停止であることとリソースへの中断のないアクセスは重要な利点である。

4) 性能

物理的な分離は、ネットワーク交信量の増大と複雑なプロトコルスタックによるオーバーヘッドという避け難い現象を引き起こす。リソースの同時利用を避け、利用可能なリソースを使うことから効率が上がり、性能が向上するという優位性を持っている。分散環境において、すべてのリソースにオンラインアクセスできるということで、そのシステムの性能が劇的に向上する。

(4)分散処理環境のフレームワーク

オープンな分散処理環境を実現するために必要となるコンポーネントを図3-4-2に示す。

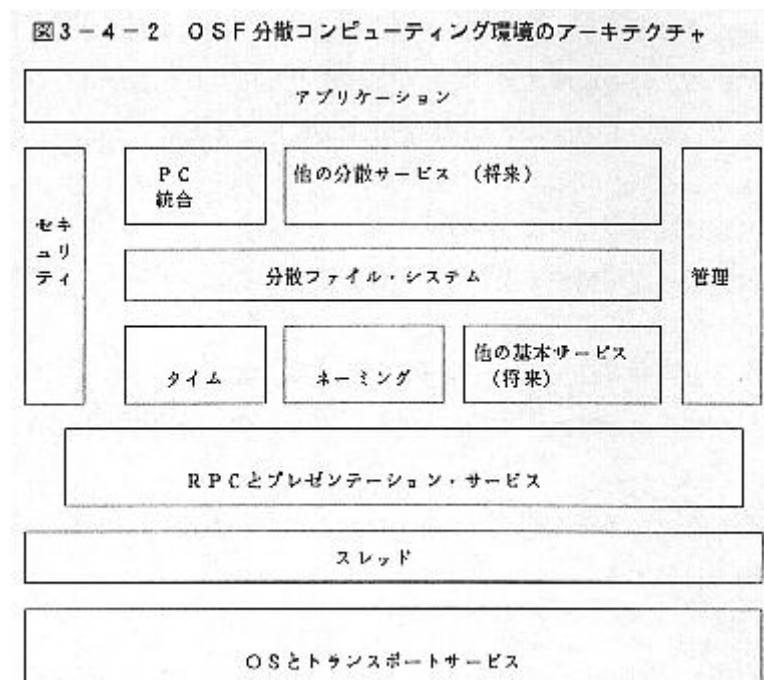


図3-4-2

コンピュータユーザーはエンドユーザー、システム管理者がネットワークの複雑さを意識せず、情報が格納されているところから必要とする情報を送れるような通信環境を要求している。

OSFのDCEアーキテクチャはネットワーク環境の物理的な複雑さを隠し論理的に単純な階層を提供する。

① 標準化の動向

1) 分散セキュリティ機能

ソリューションの多くはDCEが基本になっている。製品化も業界団体仕様が有力である。

2) 分散バックアップ／リストア機能

半分のメーカーしか計画を持っていない。個別のソリューションは存在するが明確な標準がない。

3) 分散システム／ネットワーク管理

OSI標準がすでにあるが、多くの製品化が計画されている。

4) オブジェクト管理

OMG(オブジェクト・マネジメント・グループ)がCORBA(オブジェクト・リクエスト・共通ブローカ・アーキテクチャ)を提供している

5) 分散データベース

分散データベースについてはオープンな分散処理環境で標準となる仕様が決まっていなことが問題である。ISO(国際標準化機構)がOSIの一部として RDA(リモート・データベース・アクセス)プロトコルを制定しようとしているのが現状で、IBMが独自に仕様を定め公開している。

(5) オープンな分散処理環境のシステム構築する場合の留意点

① 早い技術革新

あまりに早い技術革新で、ユーザーからみるとハードウェア、ソフトウェアに何を選択すべきか、いつ入すべきか判断に苦しむところである。したがって、ハードウェアにしるソフトウェアにしる、使用実績が多く、一般ユーザーの評価が高く、サポート体制がしっかりしている製品を選択することである。

② 非常に多くのベンダーの参加

自社システムのニーズを最適化をさせようとする多ベンダー化することになり、多くの製品群の中から自社に最適な製品を選択するため、卓越した選択眼が必要である。このために、

- ・ 日頃から新製品情報を収集しておくこと、
- ・ 収集した製品情報から自社ニーズを満足させる製品を選ぶ評価能力を養っておくことが必要になる。
- ・ マルチベンダー製品は特にソフトウェア製品に対して、他社のソフトウェア製品との間の整合性まで考慮していない。例え結合できたとしても、十分な性能が出ないとか、ハードウェアリソースを喰うとかがある。

③ 不安定、不十分なベンダーの支援体制

経営基盤の弱い新規ベンダーが参加している場合もあり、ある日突然、他ベンダーに買収されていたという話もあり、マルチベンダー製品で特に全国展開などを行う場合、各ベンダーのサポート体制等、十分な事前の調査が必要である。

④ 不完全なオープン性

オープン性についても百パーセント保証しているわけではない。メインフレームに比してオープン性が図られている程度である。

⑤ 新しい設計・管理手法の確立

オープンシステム化により、これまでにない新しい分散型システムの設計方式や、各所に分散したハードウェア機器、ソフトウェアを管理するための運用管理方式を確立する必要がある。

(6) オープンな分散処理環境を進めるにあたっての課題と提言

① マルチベンダー・オープン環境における相互運用性についてはデモ環境等で実現している事例が出てきている。ユーザー側にも確認しようという動きがある。

一方、ユーザーの立場からすると、ユーザーはまだマルチベンダー環境を取りまとめている実力を備えていない。したがって、ユーザー側の保守体制や障害時の切り分け等が不十分である。ユーザーの側に立ってマルチベンダーを取りまとめていくSIベンダーが必要である。

② 運用支援系機能の未成熟

分散処理方式の運用は従来の重装備かつ専門のオペレータによる集中管理方式から、無防備でユーザーの現場にはシステム運営の専門家を配置するのは困難であり、故障した場合の耐障害性機能、配置されたコンピュータ群を運用するための各種運用支援機能が極めて未成熟である。

③ パッケージソフトの不足

標準化に準じたツール系は多いがアプリケーション系が少ないのが実状である。

④ ユーザーインタフェースの実態

GUIはOSFグループのMotif系とUIグループのOpen Lookがあり、Motif系に統合される見通しであるが、すでに開発されたソフト資産は存在する。すでに導入したユーザーはサポートについて心配になる。システム構築をやって初めて製品の差がわかる実状であり、事前の調査とユーザーがプロトタイピングの試行で開発工程を考える必要がある。

⑤ 最大の障害はプラットフォームの違いにある。汎用機上にある既存資産とUNIX

上に新たに構築するアプリケーションやデータとを分散処理環境で共存させ連携させる相互運用性が要求される。

分散システムという新しい基盤への移行に取り組む場合のコスト増について、開発コストを抑えるためのベンダーやユーザー自身が開発に関する社内標準やガイドラインの作成が必要である。

⑥ 分散処理環境の中でネットワーク、特に伝送路としてLAN管理の問題がある。

NOS (NETWORK・OS)は伝送路に依存しない半面、伝送路の監視機能は弱い。「ケーブルの障害、WANと接続した場合のトラフィックが異常に混んでいないか、ブリッジやルーターは正常か」等の監視機能の強化が必要である。

[参考文献]

- ・ X/OPEN、標準化セミナー資料Xtra'91-92
- ・ OSF DCE 分散コンピューティング環境 (1)(5)その他

3.5 ネットワーク管理、分散システム管理

ネットワーク管理機能については、大規模ネットワークを利用した情報システムを運用していく上で必要不可欠なものとして装備され、各ユーザーにおいて利用されている現状にあ

る。近年、ネットワーク管理についてはISO(国際標準化機構)において標準化作業 が推進されている。一方、最近は情報システム技術の進展に伴い、分散処理システムが従来のホストコンピュータを中心としたシステムに代わり展開されつつあり、このシステムを運用していく上で分散システムの管理が重要な問題として浮上してきた。

このような状況の中で、ユーザーは情報システムを効率的に運用していくために必要な機能について、標準化規約に基づいた製品を利用して推進していく方法を望んでいると考えられる。

ここでISOやその他の標準化団体で検討ならびに制定されている規約、規格は、ユーザーが実際に必要としている運用機能レベルまで到達しておらず、現状ではネットワーク管理が中心となっていると思われる。

よって、ここではユーザーが実際に必要と思われる機能と、現状の標準化範囲内容、これらの比較検討を行い、抱える問題点を抽出し、さらにこれの対応について記述することとする。

(1)ユーザーがネットワーク管理、分散システム管理に求める機能

近年、展開されつつある分散型システムの管理は、分散システムのベースがUNIX機ということから、ネットワーク管理を実施することから始まったと考えられる。ネットワーク管理の内容としては、ネットワークに接続されたコンピュータやネットワーク機器などのノードの構成、障害、性能、課金、セキュリティなどの管理である。次に利用範囲が拡大していくとノードとしての管理だけでなく、システムを構成する多くの資源を集中管理したいとのニーズが出てきた。このため、コンピュータのCPU、メモリ、ディスクなどの構成、障害、性能や、サーバーから各クライアントに配信するプログラム、データなどの管理が必要となった。さらにUNIX機のCPUや入出力バス等のハードウェアの性能が大幅に向上し、RDBMSや TPモニタなど基盤ソフトも充実してきたことから、バッチジョブの運用やコンピュータの自動運転、帳票の制御、データのバックアップ等の管理機能が求められるようになってきた。

整理すると、

①ネットワーク管理

- ・構成管理
- ・障害管理
- ・性能管理
- ・課金管理

②システム管理

- ・構成管理
- ・障害管理
- ・性能管理
- ・課金管理
- ・セキュリティ管理
- ・資源管理

③システム運用管理

- ・バッチジョブ管理
- ・自動運転
- ・帳票出力制御
- ・データバックアップ

・ソフトウェア配布

となる。

(2) 標準化動向

現在、標準化されたもの、または標準化推進中のものとしてはOSI管理、SNMPとOSF/DMEがある。

① OSI管理

1) 標準化の状況

OSI管理はネットワークを管理するために機能とプロトコルを定めている。'78から検討が開始され、基本標準としては、

- ・管理フレームワーク
- ・システム管理概要
- ・管理情報構造(SMI)
- ・共通管理情報サービス(CMIS)
- ・共通管理情報プロトコル(CMIP)
- ・システム管理機能(SMF)

が検討されており、このうちSMIの6項目中3項目、SMFの24項目中14項目がまだ検討中であるが、その他の項目については国際標準(IS)となっている。

しかしながら、基本標準は考えられる機能やそれを実現するためのパラメータを網羅して定義しているため、実装レベルの標準仕様としては幅がありすぎる、または機能が大きすぎる等の問題がある。そのため実装レベルの標準仕様としての機能やパラメータを絞り込んだ機能標準が必要となってくる。これを国際標準プロトコル(ISP)と呼び、開発が進められている。ISPは一部実装規約として世の中に出ているが、必要なものがそろうまではまだ時間がかかる見込みである。

このようなことから、一部メーカーよりOSI管理をサポートしている製品が出荷されているが、ISP等の標準化が遅れているためにベンダーごとのインプリメンテーション仕様が出てくるため、ベンダーを超えた製品間の接続は実際にはできないのが現状である。

2) OSIの機能

OSI管理で実現できる機能としては、

- a 構成管理機能
 - ・ネットワーク全体の構成、及び構成要素(装置等)の状態監視
- b 障害管理機能
 - ・障害の検出・復旧・原因の特定
- c 性能管理機能
 - ・ネットワーク構成要素や伝送状のトラフィック、稼働率などの把握
- d セキュリティ管理機能
 - ・ネットワーク構成要素に対する不正アクセスの監視、保護
- e アカウント管理機能
 - ・ネットワーク構成要素の使用状況の管理

とまとめることができる。そしてこの機能を利用すればネットワーク管理は十分に満足する管理が可能となると思われる。

② SNMP (Simple Network Management Protocol)

マルチベンダー環境の中で利用できるTCP/IPネットワークの標準プロトコルであるSNMPは、OSI管理のCMIS/CMIP (Common Management Information Service / Protocol) の標準化が進むまでの短期解として開発された。

1) 標準化の状況

SNMPは1970年頃DARPA(Defense Advanced Research Project Agency:米国防総省高等研究計画局)によるInternet基本構想にさかのぼれる。

Internetは米国の研究者用ネットワークで、初期の実験ネットワークARPANET (Advanced Research Project Agency Network)によるパケット交換網を通して発展した。当初からマルチベンダー・ネットワークであり、83年以降TCP/IPを実装した。ここでInternetがあまりにも急激に拡大したため、この巨大ネットワークを管理する必要性が生じてきた。ここで考え出されたのがSGMP(Simple Gateway Monitoring Protocol)である。そしてSNMPはこのプロトコルをベースに開発された。

88年にIBA(Internet Architecture Board:TCP/IP関連のプロトコル開発を進めている米国の組織)はTCP/IPネットワークにおける管理標準設定の指針を示し、標準管理プロトコルとしてSNMPとTCP/IP上でCIMP(OSI管理プロトコル)を実現したCMOT(CMIP over TCP/IP)を認定した。以降管理対象の集合としてのMIB(Management Information Base)などを含め、RFC(Request For Comments)として短期的/長期的な視野に立った改定作業が続けられている。

IABは当初SNMPを短期間だけ使用する管理プロトコルとして位置づけ、長期的にはCMIS/CMIPのOSI管理の利用を目指していた。このため移行プロトコルとしてのCMOTも認定していたが、実装上の理由から90年8月にSNMPを当面の標準プロトコルとして位置付ける発表をした。

2) SNMPの機能

基本的にOSI管理を指向して作成されているためにOSI管理が備える前出の機能を実装すべく作られたが、その中で、

- a 構成管理
- b 障害管理
- c 性能管理

は充実しているが、

- d セキュリティ管理
- e アカウント管理

は不十分である。

(3)OSI管理とSNMP比較

OSI管理はネットワーク管理については非常に充実しているため、OSIに対してのSNMPの長所、短所としてまとめてみると以下のようなになる。

① SNMPの長所

- 1) 管理機能の数と複雑さを最小限にとどめているため、エージェントプロセスの開発が容易
- 2) 最小のプロトコルスタックで構成されているため実装が容易で、実装システム及びネットワーク管理の負荷が軽い
- 3) プロトコル自身のオプションがなく、基本的相互接続が保証されている
- 4) MIBオブジェクトの拡張ができ、管理対象となる機器ベンダーを限りなく増やせる
- 5) IPX等TCP/IP以外でも実装されはじめている

② SNMPの短所

- 1) ポーリングベースであるため、トラフィックを不用意に増大させる可能性がある

- 2) PUD(Protocol Data Unit)が少ない
単純な反面、エージェントに対する直接的な動作指定ができない
エージェント側で事象が起きたことを知らせるTrapの規定も少ない(7種類)
- 3) マネジャー間の通信ができない
 - ・ 複数マネジャーを統合管理できない
- 4) UDP(User Data Protocol)を使用しているため、認証機能やデータの保証が弱い
- 5) ベンダーの拡張MIBが増加する一方である
 - ・ 類似した機器ごとのMIB集約化が必要となる

さらにSNMPを移行用プロトコルとして規定されたCMOTと比較してみると、CMOTは37種類とコマンドの種類が豊富である。さらに、被管理対象としての動作要求、例えば障害の起きたポートを切り離すなどの能動的な管理ができる。

このように機能が豊富なCMOPであるが、現状ではOSI管理の標準化が終了してもSNMPがなくなることは考えづらいため、両者が共存していくと思われる。

(4)統合運用管理環境

2節、3節に記述したとおり、現状における標準化動向は、まだネットワーク管理に特定した管理に限られている。

1節に記述したとおり、ユーザーの特にシステム管理者はそのネットワーク管理の上位にさらにシステム管理、システム運用の機能を必要としている。

つまり図3-5-1に示すようになり、まだまだ一般ユーザーが標準に基づく製品でシステムを管理するには時間がかかると思われる。

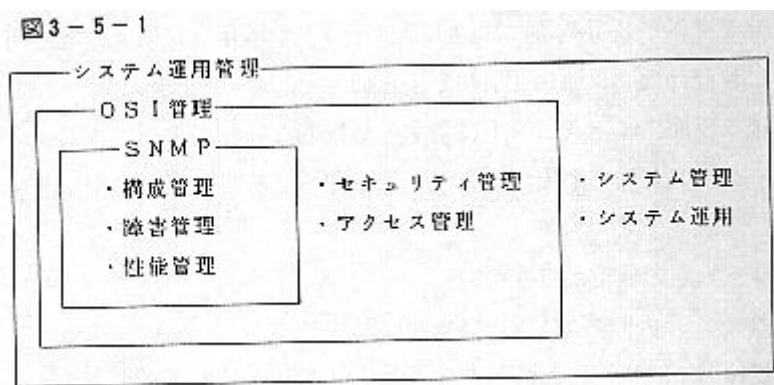


図3-5-1

しかしながら、一部メーカー等では管理するための製品として、

- ①GUI
- ②データ収集に使う通信プロトコル
- ③収集したデータを管理するデータベースのツール

などから、利用するときの共通APIを規定しようとする動きが出ている。

例えばHP社の Open View 分散管理プラットフォーム(個々のツールが共通APIを通じて利用できるGUI、通信プロトコル、データベース等を提供するソフト群)やSUN社の SunNet Manager 等がある。これらは他社にもAPIを公開して品ぞろえも強化する方針を打ち出している。

さらに、コンピュータ業界全体でプラットフォームを共通化しようという動きがある。

例えば、OSFが開発しているOSF/DME(分散管理環境)が注目を集めている。

以上のような、統合運用環境が充実してくるならば、OSI管理やSNMPが機能として取り込めなかったシステム管理やシステム運用も近い将来標準化された製品を利用できるようになると思われる。

分散システム管理についてはネットワーク管理から徐々に標準化が進められ、最近はシステム運用管理までの標準化方策が発表されるまでに至ってきたが、まだまだ製品ベースで出てくるものは少ない。

まだ、このような製品は開発しても爆発的に売れる製品とは考えにくいいためメーカーも開発意欲がないと考えられることから、ユーザーとしては、例えば図3-5-2のような共通APIの部分で標準化が進むように、これら製品に対するニーズ等をメーカー(ベンダー)に提言し、推進していくべきと考える。

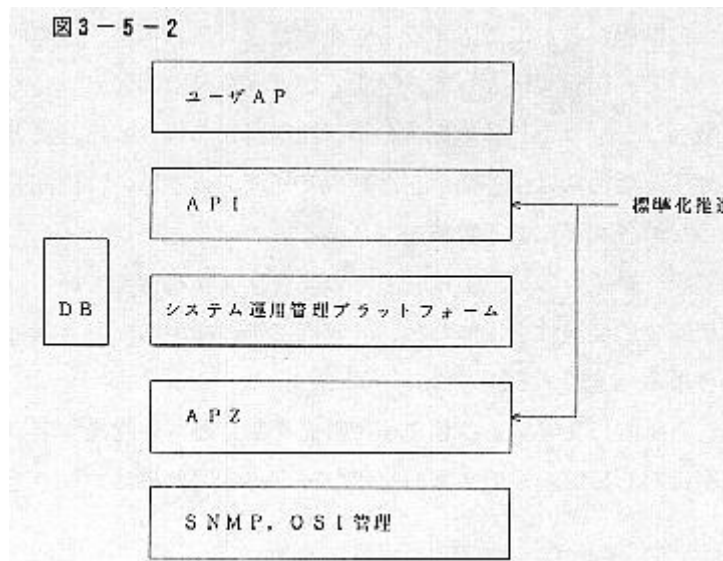


図3-5-2

3.6 文字コードセット

以下の報告では文字コードセットの概要を把握し、問題点を明確にするため、まず、文字コードセットとはどんなものであるかを解説している。解説にあたっては、技術的な完全性を追求することよりは、わかりやすいよう記述することを第一とし、技術的な専門家の方にとっては「不満を感じられる方」もいらっしゃると思うが、専門用語の使用、アーキテクチャの不完全な紹介については、目をつむっていただきたい。

これは、ユーザーが文字コードセットの変更で、どんな影響を受けるか、影響を最低限に抑えるためには、どんな対策をとっておけばよいか、対策が容易になるようなベンダーのサポートへの要望等を検討することが、本報告を作成する主目的であり、文字コードのアーキテクチャそのものを紹介するためのものではないことをご理解いただきたい。

したがって本報告の最初の部分(6節まで)は、7節以降を理解しやすくする目的だけで作成しており、読み物として参照していただきたい。

(1) 図形文字集合と文字コードセット

「田」「中」「A」「B」等の文字の集まりを図形文字集合という。文字の形を「Glyph(グリフ)」という。グリフは文字の形を問題にするもので、国内では「書体」「異体字」の問題がこれに対応すると考えられる(これらを、国際的にグリフと呼び、漢字について、国際規格として検討するかどうかは別問題)。

文字を計算機(ワープロ、PC等を含む。情報機器類を代表する言葉として以下使用する)で処理するためには、各文字を計算機で処理できる形の、内部表現とする必要がある。このため各文字にある数値を割り振る。

情報処理の対象、結果等のデータは、多くの場合、文字を含んでいる。このデータの異なる計算機間での交換性を保つため、計算機で処理をする同じ文字に対しては、同じ数値を割り振るべきである。

このように、ある図形文字集合の各文字に数値を割り振った物を文字コードセットという(一文字に対しては、その文字の文字コードあるいは単にコードと呼ぶ)。

(2)ISO646

国際規格として最も有名な文字コードセットがISO646である。幾多の変遷はあったが、現在のISO646は、米国標準で、国際的に使用されているASCII文字コードセットと同じものとなっている(旧版では異なるところがあったし、現在のJISでは、「\$」記号を使用するより「¥」記号を使用する確率が高いとして、「¥」記号を「\$」と同じコードとしている。その他にも、現JIS規格は最新の国際規格ISO646=ASCIIと一部異なっている)。

ISO646は7ビットで文字を表現している。ただし16進(以下#印を頭につけて表現する)で、#0X、#1X(X=0~F)の32個は制御コードのための領域として使用している。この制御コードの領域を、C0集合と呼んでいる。したがって、文字は#20~#7Fの96字となるが、この場合を96文字セットと呼び、#20と#7Fをリザーブし94文字のみを使用する場合もあり、これを94文字セットと呼ぶ(図3-6-1)。

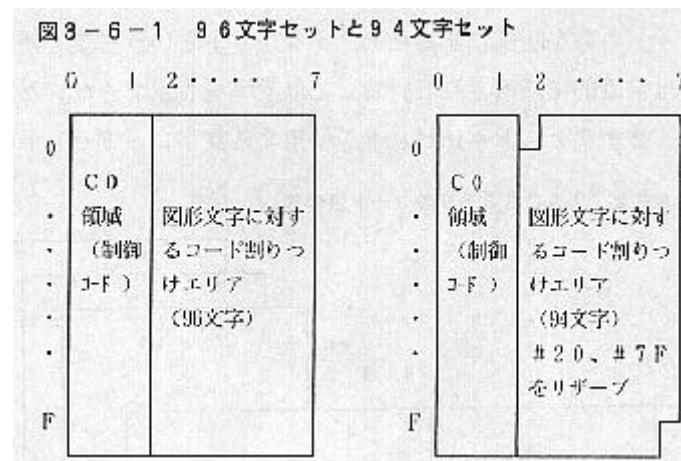


図3-6-1

(3)ISO2022、ISO8859

ところで、96文字(または94文字)では、取り扱える文字が少ないということから、文字コードの拡張方法が議論され、ISO2022として既に国際規格化されている。日本のJIS文字コードセットはこのISO2022に従って拡張されたものである。

ISO2022の拡張方法の完全な紹介は、技術的すぎると考えるので省略させていただく。

最も簡単な図形文字集合の拡張を説明しておく。ISO646では7ビットで文字を表現しているが、これを8ビットに広げてみる。ところで、ISO646の構造を継承するため#8Xと#9Xは制御コードのための領域とする(C1集合と呼ぶ)。この拡張により192または188文字を表現することができる。8ビットの全体を使用すると本来は256文字が表現できるはずであるが、C0、C1集合の領域があるため実際にはもっと少ない文字しか表現できないこととなっている。この種の拡張を行った文字コードセットはISO8859という複数パートを持つ国際規格となっている。このとき数値の小さい方の半面(先頭ビットが0となる、#00~7Fまで)を「右半面」、数値の大きい方の半面(先頭ビットが1となる、#80~FFまで)を「左半面」と呼ぶ。ISO8859は、多パートを持つ規格書であり、各パートで1つの文字コードセットを規格化している。一般に、左反面にASCIIコードを置き、右反面に欧州に必要な文字をおいたり、ラテン文字を置いたりしている(図3-6-2)。

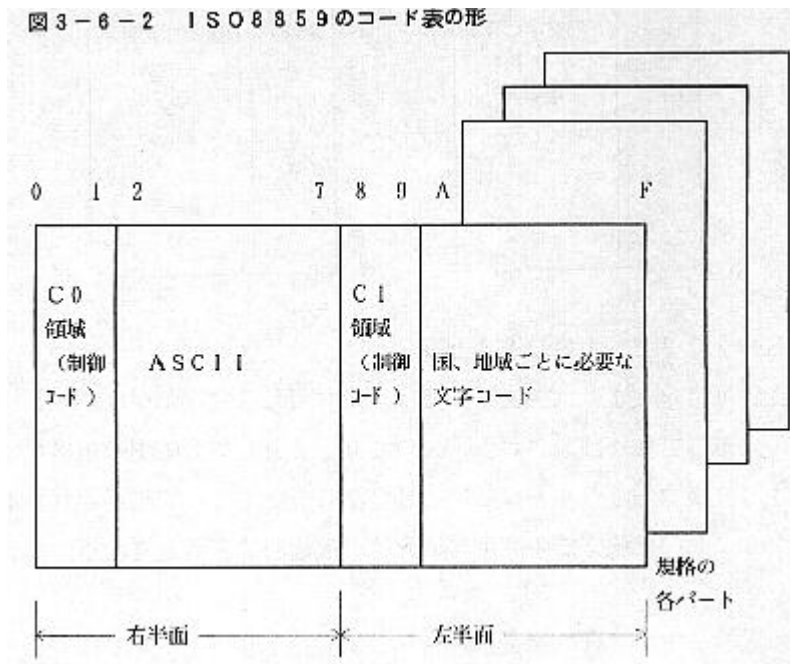


図3-6-2

日本では1オクテット(1オクテット=8ビット。バイトという言葉は、処理の単位であり、旧世代の計算機では、1バイト=4ビット、6ビット、9ビットのものがあつたため、ビットの数を明確にするため、オクテットという言葉を使い始めている。通信の規格等(では常識的に使用されている)では漢字を表現できないため、ISO2022に従って、2オクテット=16ビットを使用する文字コードセットに拡張しているが、上位、及び下位の8ビットにC0集合とC1集合に対応する数値を使用していない。このため、16ビットでは本来65536個の文字が表せるはずであるが、7ビットベースとすると $94 \times 94 = 8836$ 、8ビットベースとして $188 \times 188 = 35344$ 文字しか表現できないこととなる(96文字セット、192文字セットでも同様に計算できる)。

(4)文字コードの表現可能な数の増加について

1オクテットでは、全世界の図形文字をコード化できないことは明らかである。また2オクテットとしても、ISO2022に従う限りその図形文字の増加には制限がつくため(実際はコード表の切替え機能がありこのような制限は付かないが、2オクテット固定で 使うことを考えるとこのような制限が付く)、C0、C1集合を使用しないという制限をなくし、すべての数値を制御符号及び図形文字のコード化用の数値として使用するという考え方が現れた(制御符号 については、他に国際規格がすでにある)。

このような考え方をさして、すべてのビットの組み合わせを使用することから、「ピュアバイナリ」と呼んでいた。ピュアバイナリでは、制御符号を除いて、6万5千個以上の図形文字に対してコードの割り付けを行うことができる(図3-6-3)。

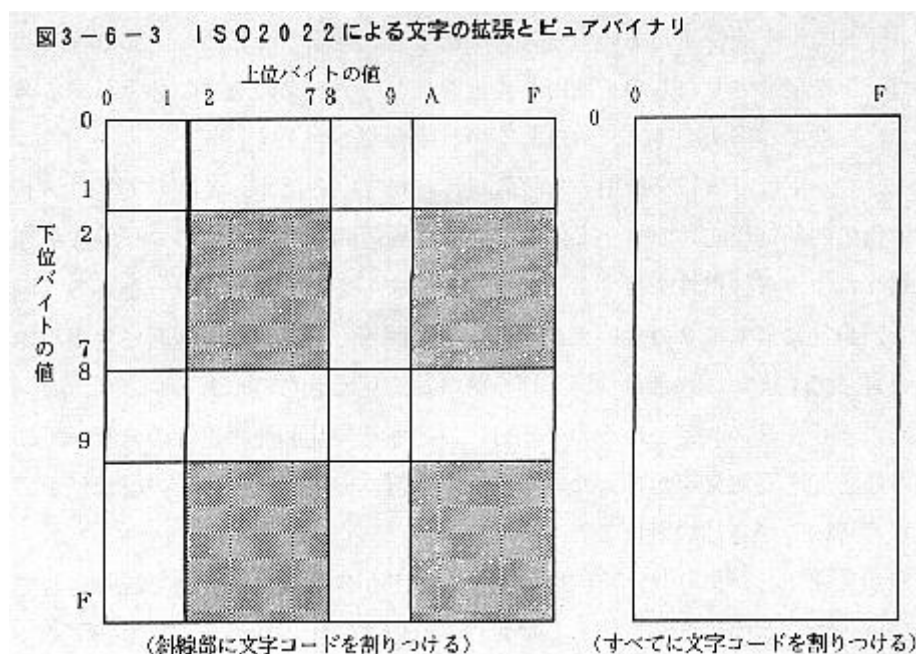


図3-6-3

6～7年前と記憶しているが、今後の文字コードセットの拡張の国際規格化をISO2022をベースとするか、ピュアバイナリとするかの国際投票を行った。この時は、ISO2022に従って、文字コードの拡張を行うことを第1プライオリティとし、ピュアバイナリを次のプライオリティとするという結果となった。

(5)文字コードセットの国際規格化作業について

図形文字集合の選択は、各国の事情があること、他の国ではよくわからないところがあるという考えから、図形文字を選択し集合を作ること、及び文字コードセットの規格化は「登録制度」(登録機関にこう決めましたと登録を行う制度で、制度の運営の仕方とも国際規格化さ

れている)で国際規格として認めることとなっていた。JISの第1水準、第2水準は既に国際的に登録されており、各国とも情報の入手は可能となっている(欧州のECMAが文字コードについての国際登録機関である)。

(6) 情報処理における日本の文字の取り扱い

日本のコンピュータメーカー、ワープロメーカーは、以上に述べてきた国際規格に基本的には従いながらも、ISO646とJIS漢字コードの共存、半角、全角の取り扱い等を含め、きめの細かいサポートを行ってきた。なお具体的な計算機の実装では、JISコードを使用していると言いながらも、実際の計算機の内部処理では、JISコードを変形させたものを使用し、他機種とのデータの交換の必要がある場合に、JISに変換する方法を持ち、対処している場合が多い。

JISコードは「情報交換用」と位置付けられているため、各社の文字コード体系(計算機の内部処理向けのコード体系や各社の周辺機器への出力のコード体系等)に、上に述べたような独自性を持つことになってしまっている(メーカーからの立場で考えると、自社に最も都合のよいようにJISを解釈し、許される範囲で変更を加えていると言ってもよい。許されないという厳格な意見はもちろんある)。

さらに、JISの改定と、そのJISに対応する製品開発の追従のスピードの相違、JISに追加・変更文字があった場合の旧資産(ユーザーデータ)の継続使用の保証の方法の違い等が各社の事情で異なっている。

これが、JISがありながら、標準文字コードが各社間の計算機で統一されていないという一つの原因になっている。なお、これ以外に、各社がJISコードセットの未使用領域に勝手に固有の文字を追加しているため、この部分も各社固有となり、各社の計算機間の文字データの交換の妨げになっている。

さらに、処理の複雑さを減少させるための工夫を施して開発されたシフトJIS、ISO2022に従ってUNIXのために開発した文字コードセット(EUC:Extended UNIX Code)等がサポートされることになったため等、各種の要素が複雑に絡み合っ、日本のメーカーの文字コードセットの計算機によるサポートは複雑なものとなっている。同じ会社の中でも、メインフレーム、ワークステーション、PC、ワープロで、それぞれ違ったコード系をサポートしている場合もあるし、極端な場合には、同じレベルの計算機でも文字のサポート範囲が異なり、交換ができないという場合すらある(これと同じような課題として、媒体の非互換の問題もあるがここでは触れないこととする)。

それぞれのシステムの規模によつての差、利用方法の差等からこのようなケースが生ずる場合があるが、一般には、同一社内ではコード変換のツールを準備している。ただし、他社の機種間では変換が困難な場合がある。この最大の原因は、規格に含まれる文字数が少

ないため、各社独自に追加文字を規定しており、これらが日本全体としては統一されていないことになっている。

さらに、利用者固有の文字の登録を可能としている場合があり、これらをメーカーが使用していない領域に割り付けると、異なる社の計算機では空いている領域の違いから、これらの文字すら異なる文字コードを持つようなことも考えられる。

米国では、日本における8ビット、16ビットの混在した文章、データのこのような細かな処理、複雑な体系について、日本に及ばないという危機感が次第に大きくなってきた。これがエスカレートして、8ビット、16ビットあるいは32ビットで表される文字の混在した文章、データの処理が必要となる中国(台湾、香港、及びベトナム等の中国人が多い東南アジアの国等を含む)、韓国、タイ、中近東、エジプトその他多数の国での米国メーカーの地位が危なくなるのではないかと危機感が広がり(のではないかと私は想像している)、文字のできるだけ統一した、わかりやすい拡張の検討が行われてきた。

(7)7ビットコードから8ビットコードへの移行

ATTの開発したUNIXは7ビットコード(ASCII)を使用していた。このため、システムの開発者は、現在の計算機の取り扱うバイトは8ビットが主流であり、UNIXも当初から8ビットが基本の計算機上に開発されたことから、メモリ領域の有効利用を考えて、最上位の1ビットを処理の情報用に使用していた。このため、日本の多くのメーカーやIBM等を含め、文字コード体系として8ビットを使用していたメーカーにとっては、従来から使用していたコード系をUNIXでは使用できないこととなり、ATTへ要望をだし、最上位ビットを使用しないような改造を提案、ATTも了承し、第1ビット目を使用しないようにする改造の検討が行われた(図3-6-4)。

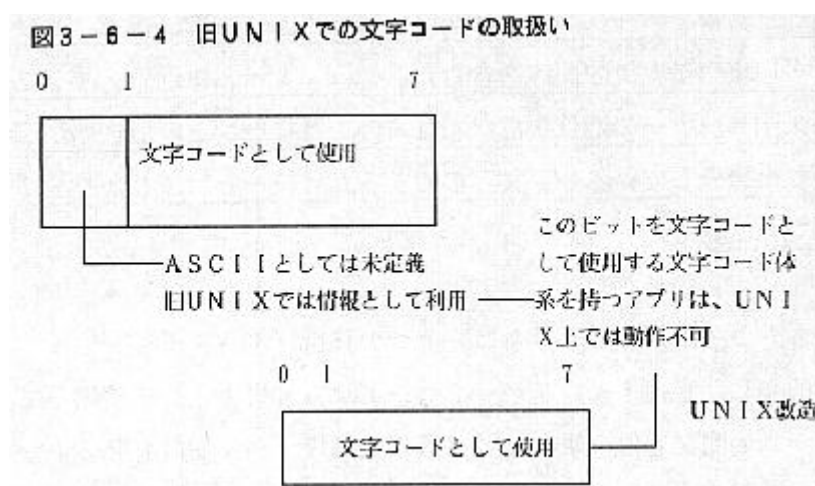


図3-6-4

この改造と同時に、日本語の取り扱いを可能とすることが、日本のメーカーとATTを中心とする米国勢で検討され、前記のEUCが開発された。

ここで大きなステップがあったと私は考えている。

すなわち、これまでのコードセットの標準化は「情報交換用符号図形文字コードセット」であったが、この検討の過程で、このほかに「内部処理用コードセット」、「ファイル用コードセット」の考え方が提唱されている(実用的には、既に多くの社が区別して使用していたが、公開される文書にこのような名前が出たのは初めてであろう)。

内部処理に適したコード系としては、例えばすべてのコードを16ビット(2オクテット)で表現することにより処理が容易となる。またファイルに保存することに適したコード系は必要領域を最少とするというように、文字コードを使用目的にあったものとする考え方である。もちろん、これらのコードの間に1:1の対応を付けられることが前提条件となる(図3-6-5)。

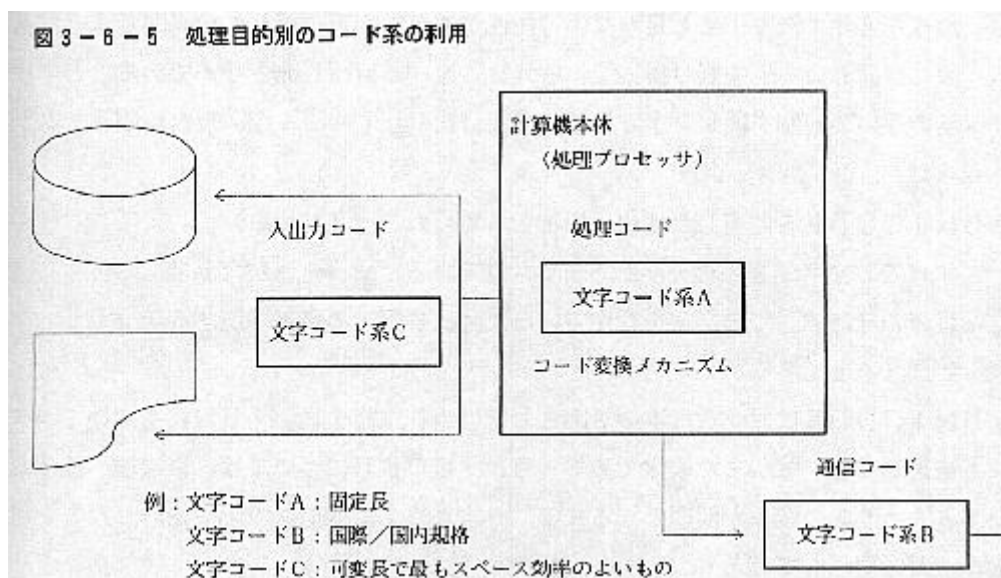


図3-6-5

以上で、ほぼ本論のISO10646及びUNICODEの出現までの話がすんだ。以下、新しいコード系であるISO10646とUNICODEの話に移る。

なおJISの具体的な話を記述しなかった。これらとしては、JIS X0201:X0208、0212があるが、これらの具体的な紹介は省略する。

(8)UNICODEとは

UNICODEは、米国のベンダーが集まって開発をした新しい文字コードセットである。検討の中心／開始したのは、Microsoft、アップル、ZEROX等であったが、現在は会社組織 UNICODE Inc. (IBM、DEC等多くの参加企業がある) として活動を行っている。

UNICODEの概要は以下のとおりである。

- ・ 2オクテット(16ビット)を使用するコード系である。

前記の言葉で言えば、ピュアバイナリコード系であり、C0、C1領域に対応するコードも自由に使用している。

- ・ 世界のほぼすべての文字を含んでいる。
- ・ 合成文字を作る機能がある(例えば、ハングルやタイ語等のように字母を持っており、それらを重打ちする〈3、4つの字母の重打ちもある〉ことで、一つの文字とする)。
- ・ 日本のKANJI、中国のHANJI、韓国のHANJAに別々のコードを割り振ると2オクテットでも足りない可能性があり、字形の同じもの／同じようなものには同じコードを割り振っている(HAN Unificationと呼んでいる。どの程度の違いの範囲を同じ文字と見るかのUnification ルールが、以前は明確でなかった)。

UNICODEを開発した理由は以下と考えられる(メリット)。

- ・ すべての文字が2オクテットで表現されているため、処理が容易である。
- ・ 世界のほぼすべての文字を含んでいるため、各国での文字に関するテーラリングがほぼ不要である。

UNICODEに対する日本の対応としては当初は富士通、IBM社を除いてコメントを出してはいなかった模様である(UNICODEについては、後に述べるISO10646との関係が問題となるまで、日本として組織的な対応を検討したことはなかったと記憶している)。

UNICODEはこのように米国内で検討が進んでいたが、ISOの新しい規格化作業として、次に述べるISO10646が開始されるに及んで、一躍国際の場にその名前を出してきた。

ISO10646の中にBMP(Basic Multilingual Plane)と呼ばれる特別な領域があるが、この領域の内容をUNICODEと同一とする案であり、これについて以下の項で述べる。

なお1994年春には、UNICODEの紹介(説明会とデモンストレーション)が日本で行われており、米国の計算機では既にUNICODEがサポートされていることをご覧になった方も多いと思う。

(9)ISO10646の検討経過について

ISO10646はISO/IEC JTC1 で検討を行った、世界中の文字コード系をひとつの体系としてまとめあげる(基本的には) 4オクテットの文字コード系である。これら4オクテットで表現できる数値の領域を、いくつかの領域(プレーンと呼ぶ) に分け、目的ごとに使用する(図3-6-6)。

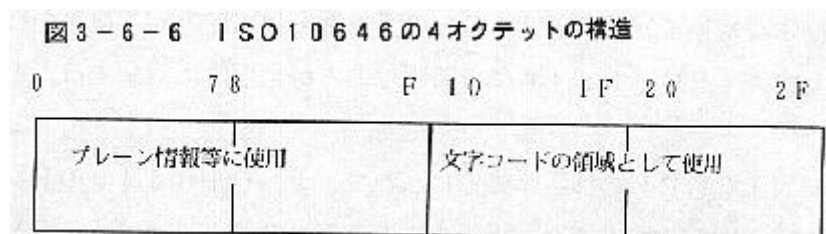


図3-6-6

コード系としてはピュアバイナリであるが、各プレーンは、もしそれが日本に割り当てられたプレーンであれば、日本が自由に検討し、使用方法を登録すればよい。

ISO10646の考え方については一般的には大きな問題はないが、あるプレーン、上述のBMPについては大きな論争が起きた。

BMPは、UNICODEの開発目的と同様、ほとんどの世界の図形文字集合を含めようとしているプレーンであり、文字コードを表すデータ(文字コードに割り振る数値)としては、4オクテットの内の、下位の2オクテットを使用している。上位の2オクテットは、これがBMPであること等、プレーンを識別したりする情報として使用される。ISO10646には、コンパクションメソッドというものがあり、ある制御のデータが表れると、それ以後のデータストリームでは、BMPプレーン内のコードの部分だけを使用するということを宣言できるようになっているため、UNICODEと同様、2オクテットで文字の処理が可能である(図3-6-7)。

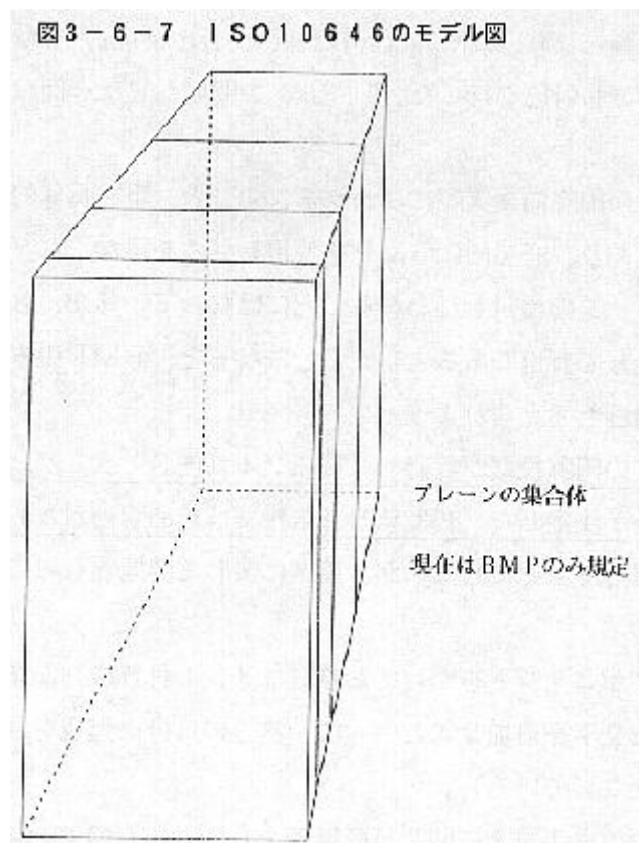


図3-6-7

当初は(簡単な変換で)現在のJISとの互換性を保てる方向で検討が行われており、日本としてもISO10646には問題はあるが全面的に対決する、とはしていなかった。

しかし、UNICODEの体系が整うにつれて、ISO10646のBMPとUNICODEという同じようなものが世の中に2体系あることはまずく、BMPとUNICODEの整合化を図ることが国際会議で決議された。

BMPに含める漢字については、先に述べたHAN Unificationが必要であり、そのとき、どんな順で漢字を並べるかの検討が行われたが、中国の「コウキ辞典」を基にすべきとの意見が強く、日本としても、その案をのまざるを得なかった。

この時点で、日本のコード系の国際対応委員会としては、JISとの互換性については検討のプライオリティを下げざるを得なくなった。ただし各国際会議ごとに各国の固有の(しかし国際的には登録された正規の)コードがあり、既存資産との互換性の保持の重要性については問題であるとしてコメントしてきた。

また実際のHAN Unificationの作業については、China、Japan、Koreaで解決すべきとの国際的な結論となり、CJK-JRG(Joint Research Group)を結成し、日本としても対応委員会を設置し対応を行った。CJK-JRGの問題などは今回は省略させていただく。

ISO10646が国際的に実際に使われるとすると、現在具体的に決まっているものはBMPのみであり、必然的にBMPを使用せざるを得なくなる。

さらに、他のプレーンの検討も行われることにはなっているが、BMPが本当に、Multilingual処理として有用であるということになると、BMPのみがISO1646の中身となる可能性も考えられよう。

ISO10646の国際投票が行われ、賛成24(内P-メンバーは20) 反対6で国際規格となることが決定した(1992年5月末投票は締め切られた)。

いまさら愚痴を言っても始まらないが、投票に関して問題となることが2点あったと考える。

1つは、多くの国がコメントをつけており(コメント付賛成16か国)、その中で、BMPへ新たな図形文字を追加せよといっているものはほぼ無視をされた(次のバージョンの検討課題となっている)。

他は、JTC1の投票手続きに関しての無知から生じたもので、コメント付き賛成だが、コメントの中にこれが受け入れられないと反対にまわるというものがあることや、資料は別途送付するとしていることである。JTC1の規定では、賛成投票はコメントがどのように取り扱われようとも「賛成」であり、「反対」にまわることはできず、意見を通す必要がある場合は、「反対(この場合必ずコメントが必要)」としなければならない。また後者の問題は、国際投票に関する情報については、委員会に直接持ち込むことは正規には認められていず、必ずJTC1のITTF(Information Technical Task Force: JTC1の中央事務局に当たると考えればよい)を通す必要があるため、ある程度の時間的余裕をもって送付する必要があるのだが、これらの規定を知らずに投票をしているとしか考えられない。

さらに多くのコメントを読むと、BMPを作ってくれば自国のためになると単純に考えている国が多いようだ(自国ではコンピュータを作っていないし、まだそれほどコンピュータライゼーションが進んでおらず、データ互換等問題ではないと考えている国がかなり多い)。

(10)日本としてのISO10646 BMPの問題点

いくつかの問題点を日本は指摘している。

しかし、日本単独のコメントはほとんど受け入れられていないといってもよいと考える。

本日はテクニカルな状況は割愛させていただくが、最大の問題は既存国際規格(登録された仕様を含む)との互換性の確保である。

互換性確保の方法は考えられるが、どのような方法で行うべきかの技術的な検討は進んでいない。

現在欠如している情報は、実装システムとしてISO1646 BMPまたはUNICODEを使用する所があるのか、あるとして、どのような実装を考えているのかであり、実装の方法によってはインパクトを最小に押さえられる可能性もある。もし情報交換用コードとしてのみの使用であれば、コード系が1つ増えただけ(これも大きなインパクトではあるのだが)、内部処理コードとして使用するときはその実装方法で大きく互換の取り方が変わってくる。したがって、日本も情報交換用コードとしてISO1046の BMPは使用できますと言うだけでよいのであれば、費用はかかるがユーザーへの影響はほとんどないはずである。

日本としては、BMPについては問題があるとしているが(国内で広く利用されている文字、記号類がない。反対に、ほとんど使われていない記号が日本オリジンとしてある等)、それらは今後修正していくこととし、BMPの良さに目を向け、また既に述べたように、UNICODEの実装者が現れつつあることを考え、ISO10646のJIS化を行うことを決定した。JISの原案は、まず翻訳、JISとの対応表を作成するなどして、JISとしては初めてと考えられるパブリックレビューを行い、コメントを広く一般に要請し、それらのコメントを解決して、1994年3月末、JIS化委員会としての最終資料のまとめを行った状況である(JIS X0221となる予定)。

なお、先に述べたように、現在のJISとは互換性はないので、付属書としてX0201、X0208、X0212との対応表を載せている。

さらに、JISの本文ではないが、解説書、使用方法の注意事項等、JISを使用するうえでの留意事項をまとめていく計画である。

JISの案の完成からJISの制定までには少なくとも半年程度は必要であり、この秋、あるいは遅くとも1994年度内には、ISO10646に対応するJISが制定されると考えられる。

なお、日本のベンダーでISO10646を開発する社があるのかどうかは、現状では製品計画の段階と考えられ、明らかにはなっていない。しかし米国オリジンのシステムは、一般的にはUNICODE=ISO10646のBMPをサポートするであろうことは想像される。

(11)まとめ

以上、簡単に文字コードの紹介を行った。

しかし、ユーザーとしては、文字コードセットの差異を直接意識して、プログラムを作成したり、ワープロの入力を行うことは苦痛である。したがって、文字コードの差異はできるだけベンダー、ソフトウェアハウスの提供するソフトウェアで吸収していただきたい。

文字コードの取り扱いについての具体的な提案はユーザーとしては困難であるが、

- 1)上の目的を達成するためにユーザー自身が気をつけておくべきこと
- 2)使用する文字コードセットを変更する可能性を考える場合に、ユーザーが気をつけてお

くべきこと等を事前に周知できるような利用マニュアルの完備
3) もし実際に移行を行うことになった場合には、移行を助けるツールの開発
等の検討をよろしくお願いしたい。

第4章 方法論、見積り標準関連

4.1 CASEについて

CASEとは、システム開発者の設計・製造・試験活動を計算機を用いて支援するツールである。CASE はさらに、開発工程のうち要求分析工程、基本設計工程を支援するツールを上流CASE、また詳細設計、製造、試験工程を支援するツールを下流CASEと大まかに分類される。

個々のCASEツールは様々なシステム開発活動のうち一部の活動を支援するのみである。開発工程全体を支援する環境を構築するためにはこれら個々のツールを組み合わせる必要がある。これを統合CASEと呼ぶ。

システム開発の効率化は、システム開発者のみならずユーザー、経営者など多くの人の関心事である。しかしながら現状ではCASEの普及はあまり進んでいない。

本稿では、CASEの普及における問題点について述べ、CASE標準化の動向と問題点について述べる。

(1)CASEの普及における問題点

CASEの普及の阻害要因として、ユーザーの期待とCASEの約束する効果のズレを指摘する意見がある。すなわち、開發生産性についての捉えかたのズレである。ユーザーの期待するものは即効性であり、目先のバックログの解消であるのに対し、CASEの提供するものは開発全工程トータルの生産性向上であるというものである。さらに、「構造化手法が日本人になじみにくい」と考えるユーザーが多いことも指摘されている。これらは現象面としては事実であろう。これらは、CASEそのものに対するユーザーの理解が進み、構造化手法の習熟に伴い、徐々に解消されていくであろうと考えられる。むしろ、CASEの普及の阻害要因とはユーザー自身そのものの問題なのではなかろうか。CASEとは、いわばシステム開発のモデル化である。モデル化されている分だけ現在のユーザーのシステムをめぐる問題点が浮かび上がりやすい。この点がCASEの導入に二の足を踏ませるのではなかろうか。

上流CASEあるいは統合CASEには、たいてい経営戦略を対象とするフェーズがある。計画局面などと呼ばれるこのフェーズは、自社の資源・強み・弱み・成功要因・重要課題とその優先度等の定義をもとに、開発すべきシステムの対象領域と開発目的を定義することになる。このフェーズの推進主体はユーザーであり、システム部門はシステム技術の専門家としての参画が求められる。しかし、実際問題としてシステム部門もまた経営戦略についての取り組みは当然要求されよう。だが、情報システム部門を巡る様々な批判には経営問題についての取り組みのなさが挙げられることも多く、現在のシステム部門の能力・体制等から、

一朝一夕に経営戦略に関わることが可能なユーザーは多くはないと考えられる。また、トップマネジメントのシステムへの関与が欧米に比して低いこと、CASE導入の初期投資が大きく効果が出るまで時間がかかることなども、システムの戦略的位置づけを低くしている。仮にCASEを導入したとしても、これらの局面のツールの有効利用が難しいユーザーが多いと考えられる。経営者及び情報システム部門の責任者がこうした事項に理解を示すことがCASE導入成功の必須要件である。

一方、CASEツールは厳格なトップダウンとウォーターフォールモデルを採用しているものが多い。したがって、万が一要件の抜けが発覚した場合、進捗に致命的な影響を生じる場合もありえる。ベンダーによっては、構造化手法に教条的に従う必要はないという説明をしているところもあるようだが、手法の簡略化・変更の許容範囲が明確でない限り、後工程にどのような影響が生じていくのかが把握しにくいものである。開発途上における仕様変更・要件変更に対応可能なCASEの提供が強く求められる。

(2) 開発支援ツールの標準化動向

現在の標準化の方向は、ツールのインタフェース(ツールの走行プラットフォーム、データ交換形式、ツール間制御など)を統一する方向である。これはユーザーが様々なCASEツールを組み合わせる使用することを容易化するのが目的である。ツールの機能や、ツールが支援する開発方法論の標準化は行われていない。

① CMA

以前より、開発環境のフレームワークの概略としてECMAのPCTE (Portable Common Tool Environment) が知られている。このモデルには、ユーザーインタフェースサービス・プロセス管理サービス・オブジェクト管理サービス・通信サービスの4つのサービスが規定されている。ただし各サービスのAPIについての具体的な規定については完全な形のものがない。この中で特に重要なものは、オブジェクト管理サービスに属するリポジトリ(設計データ管理サービス)である。これはCASEツールで作り出す設計情報を一元管理するデータベース機能である。現在、PCTEは国際標準化をFast Track(早期標準化の手続き)により手続き中である。

② COSE

昨年3月、統一UNIXに向けてCOSEが発表され、同6月にはデスクトップ環境 (CDE=Common Desktop Environment) の最終仕様が決定している。COSEは通信機能のAPIとして、HPのSoft Benchが採用されている。ここに業界標準の有力候補が登場したことになる。COSEの目的は、APIを統一することでサードパーティーのソフト開発を容易にしアプリケーションを増やすことであり、Windows NTに対抗するものであると言われている。その

後、COSEに参加しているサンソフト社がNextstepのライセンスを受けることが発表される等、業界標準の確立には今しばらく流動的状況が続く模様である。

③ EIA

CASEツール間での設計データを交換するための言語構文としてEIAのCDIF (CASE Data Interchange Format)がある。現在これをベースとした国際標準が検討されている。

(3)標準化をめぐる問題点

以上、標準化の動向をおおまかに見てきた。これらの標準化はユーザーにとって益はあるのだろうか。全体の傾向として、ベンダーはデータ交換・共有の標準化にはあまり積極的とは言えないようである。また、データ交換・共有についての標準化は、設計データの記述言語や設計データを格納するリポジトリについての標準化が中心であり、CASEツールの作る設計データの中身についての標準化、すなわち方法論やCASEツールの持つべき機能の標準化やCASEツールが作り出す設計データが互いにどのような関係にあるのかという標準化が行われていないようである。

CASEの導入が成功し利用が進めば、膨大な設計情報が蓄積されることになり、リポジトリの有効利用がユーザーにとっての最大関心事となろう。このとき異なるCASE間でデータの互換性が保証されていなければ、ユーザーにとって様々な不利益が生じることは容易に想像がつく。こうした問題についてベンダーの積極的取り組みを期待するとともに、ユーザー自身も積極的に発言する姿勢が強く望まれる。

4.2 オープンシステム(クライアント/サーバーシステム)開発方法論

(1)現状の問題点

汎用コンピュータ系の情報システムは歴史も古く開発方法論が確立され開発の進め方や方法論について標準化が進み、社内標準として完成しているところが多い。

一方、オープン系の情報システムについては、汎用機系との違いを認識しているにもかかわらず、汎用機での開発方法論に準じて開発を進めたり、簡単な標準を作って開発を行っているのが大部分であり、方法論として体系的にまとめられたものを活用している例はほとんどない状況である。

現状の認識として、大きな問題点と思われる項目を挙げてみる。

① OSの成熟度及び機能

オープン系システム開発は従来型システム開発に比べて、開発者側の担当範囲が広い。それは、成熟した汎用機のOSに比べるとオープン系OS (UNIXなど)の機能が不足していることに起因している。

例えば、従来は極端に言えば適用業務設計のみに没頭すれば情報システムが開発できたが、オープン系システムでは、OSを適用業務に最適化して使用するための開発作業や論理上は実現可能な仕組みでも実際に稼働するかどうかプロトタイピングなどにより確認してから次のステップに進むなどの作業が、システム開発 上重要な作業として発生する。

② オープン化/ダウンサイジングシステムは分散処理が前提

ダウンサイジングにより能力の小さなコンピュータを使用することになるが、1台では能力不足の場合が多いことから分散処理、クライアント/サーバーモデルの形態をとることが多く、規模の大きいシステムでは超分散システムとなる。

そのために、LANなどのネットワーク技術、最適なデータベース設計、OLTPなどのミドルウェアの設計など、システム開発者が考慮しなければならない範囲が広く、高度なノウハウが必要である。

③ システム規模は汎用機系に比べオープン系は小規模システムであり、開発スピードが早い

汎用機系システムは集中型が多く大艦巨砲的システムが多く、システム開発もおおがかりで、大勢の開発者が参加するため開発工程も細分化されており、結果として開発工程も長いものになっている。

しかし、オープン系システムでは、ダウンサイジングされたシステムでシステムの規模が小さい場合が多く、汎用機系のシステム開発工程のように細分化された手順ではなく、大きくりの工程で効率よく、しかも抜けがないような開発工程が望まれている。

④ 利用部門がシステム開発に参画する度合いが高い

オープン系システムはより利用部門に近いシステムとなり、利用部門自らシステム開発に参画するケースが多い。

そのために開発ドキュメント、開発手法もオープン化が必要になり、従来のように情報システム部門のSEだけがわかるような専門的なドキュメントでは困る。

⑤ 利用部門でシステム運用

利用部門主導で開発されたシステムは利用部門で維持・運用を行い、現業部門の改善・改革をタイムリーにサポートしていく必要がある。

そのためには、システムの仕組みや内容、データの意味や作られ方などが利用部門で理解できなければならないが、利用部門のためのシステム公開の仕組みがない。

⑥ コンピュータサービスは共通サービスから個別サービスへ

汎用機系システムでは、全社の業務を標準化(共通化)して構築するという考え方であったが、オープンシステムの出現は部門個別の業務に対応する個別サービスの考え方で構築しなければならない。

(2)標準化の動向

① 「システム開発の共通フレームに関する委員会」からの提案

通産大臣の諮問機関である産業構造審議会情報産業部会から、ソフトウェアに関しての「マーケットメカニズム確立のための基礎条件の整備」として取引ルールの明確化が指摘されたのを受けて、情報処理振興事業協会で「システム開発の共通フレームに関する委員会」で平成4、5年度にわたって活動を行ってきた。

その成果物として

「ソフトウェアを中心としたシステムの取引に関する共通フレーム体系」、及び「ソフトウェアを中心としたシステムの取引に関する共通フレーム定義」(表4-2-1(文字が小さ過ぎるため掲載しておりません))が完成している。

内容的には契約作業、システム開発作業の標準の枠組みと手順が示されたもので、必要な項目が網羅されている。

しかし、この成果物はシステム開発やシステム開発方法論の社内標準を作成することを目的としていないため、オープン化/ダウンサイジングにおけるシステム開発方法論という最新のノウハウのために活用するには適切ではない。

② ベンダー提供開発方法論

コンピュータメーカー、システムコンサルティング会社から提供(市販)されている方法論が流通し始めている。

しかし、製品の質、量とも汎用機系に比べるとプアーであり、これから急速に品揃えが行われるものと思われる。

しかし、オープンシステムでは上流工程はともかく、下流工程(製造工程)はそれぞれの企業やシステム形態・内容により特徴があり、カスタマイズなどの作業をしないと社内標準としては活用できない。

(3)オープンシステム(クライアント/サーバーシステム)開発標準の考え方

① 基本的考え方

オープン化/ダウンサイジングシステムは分散処理(超分散)が前提になるが、システム全体の整合性をとるということが重要になってくる。

また、従来のシステム開発は個別システムがそれぞれ独立して開発され、連携が取り難いという問題を提起されることが多く、都市計画型システム開発という考え方が重要になってくる。

企業の仕組み全体を見通して、利用部門、情報システム部門の関係者全体で都市計画(システム全体構想)を作り、システム間インタフェースとしてのデータベースを固め、インフラを整備してから個別のシステム開発に取り組むというアプローチが必要である。

このような開発の進め方をする事により、個別システムは個性を發揮し、利用部門に最適なシステムが構築でき、しかも全体の整合性も取れることになる。

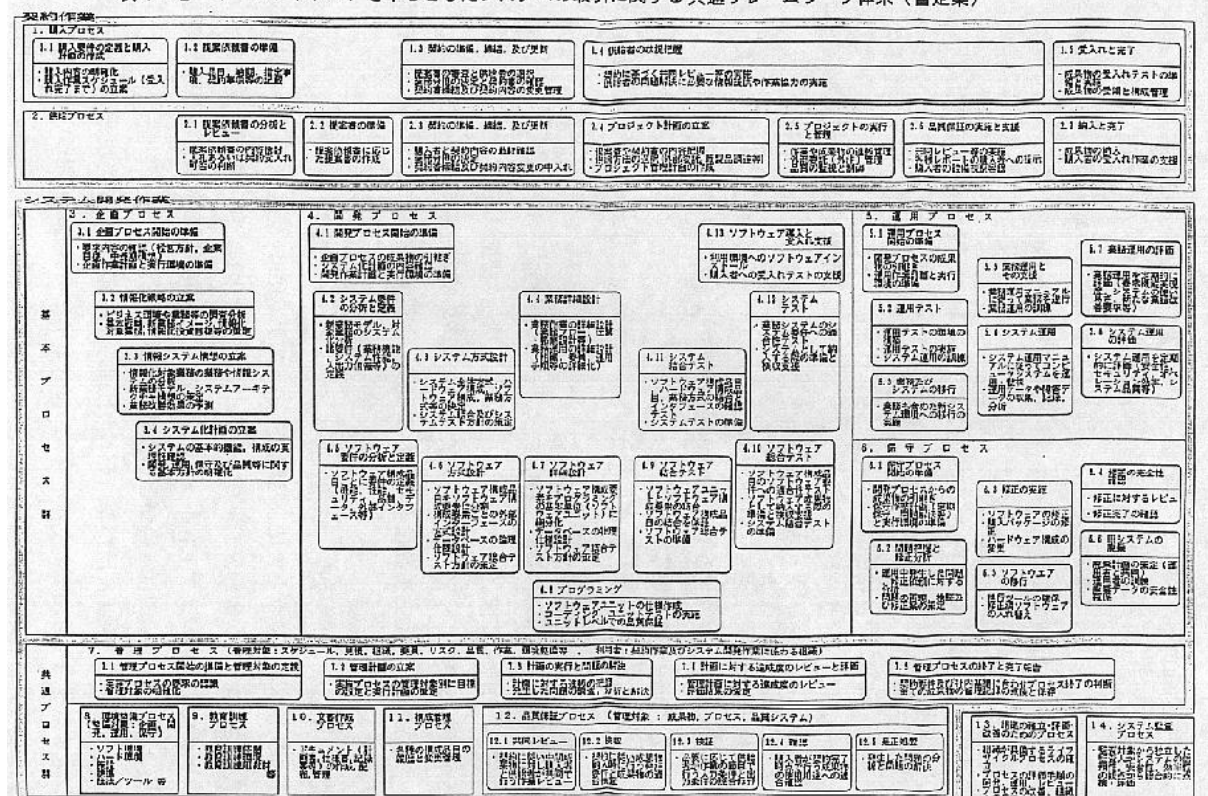
そして、利用部門は自分の要望・予算の範囲を明かにし、情報システム部門では都市計画を遵守する中で要望に応えられるように設計図(開発ドキュメント)で設計し、その設計図で利用部門とのコミュニケーション及び製造作業を行う。

製造作業は個別受注を受けてその都度生産するオーダーメイドではなく、モジュール合成で個別ニーズに対応するという基本的な考え方で実施する。どのようなモジュールにするか、既製のモジュールが再利用できるかがポイントであり、そのために開発の進め方も考慮する必要がある。

既製モジュールの再利用をすることにより、納期の短縮、個々の品質保証が全体の品質を向上させ、コストの最適化が実現できるのである。

1) システムを3つの分野に分離(図4-2-1参照)

表4-2-1 ソフトウェアを中心としたシステムの取引に関する共通フレームワーク体系(暫定案)



注1) Xに番号の名称はアクティビティであり、その下に代表的なタスクを示す。これらのプロセス/アクティビティの位置は実行順序を規定したものではない。また、必要に応じてプロセス/アクティビティ/タスクを取捨選択したり、繰り返して実行したり、順次を繰り返して実行しても良い。注2) 共通プロセス群のアクティビティは、他のプロセスから必要に応じて参照され、実行される。

図4-2-1

クライアント／サーバーモデル(C/Sモデル)が前提のシステム開発では、設計の上流工程からC/Sモデルに適した設計の進め方をしなければならない。

具体的には、情報システムを、

- a コンピュータアウトプット帳票や画面のような人と会話するマンマシンインタフェースの分野
- b データチェック、計算などコンピュータ処理するコンピュータプロセスの分野
- c データを保管、管理、検索するデータコントロールの分野

の3つに分離して設計を進めることにより、機能の明確な分離と独立性を高めることができるので、C/Sモデルの設計の進め方の基本として重要である。

このような考え方を徹底することにより、再利用可能なモジュール化、部品化が実現できる。

- 2) 開発ドキュメントは公開を前提に作成(図4-2-2参照)

図4-2-2 システム開発標準の考え方（特徴）

【開発ドキュメントは公開を前提に作成】

開発ドキュメントは

「利用部門での運用を可能にするため」「開発ドキュメントの信頼性向上」

を図るため下記2項目を規定。

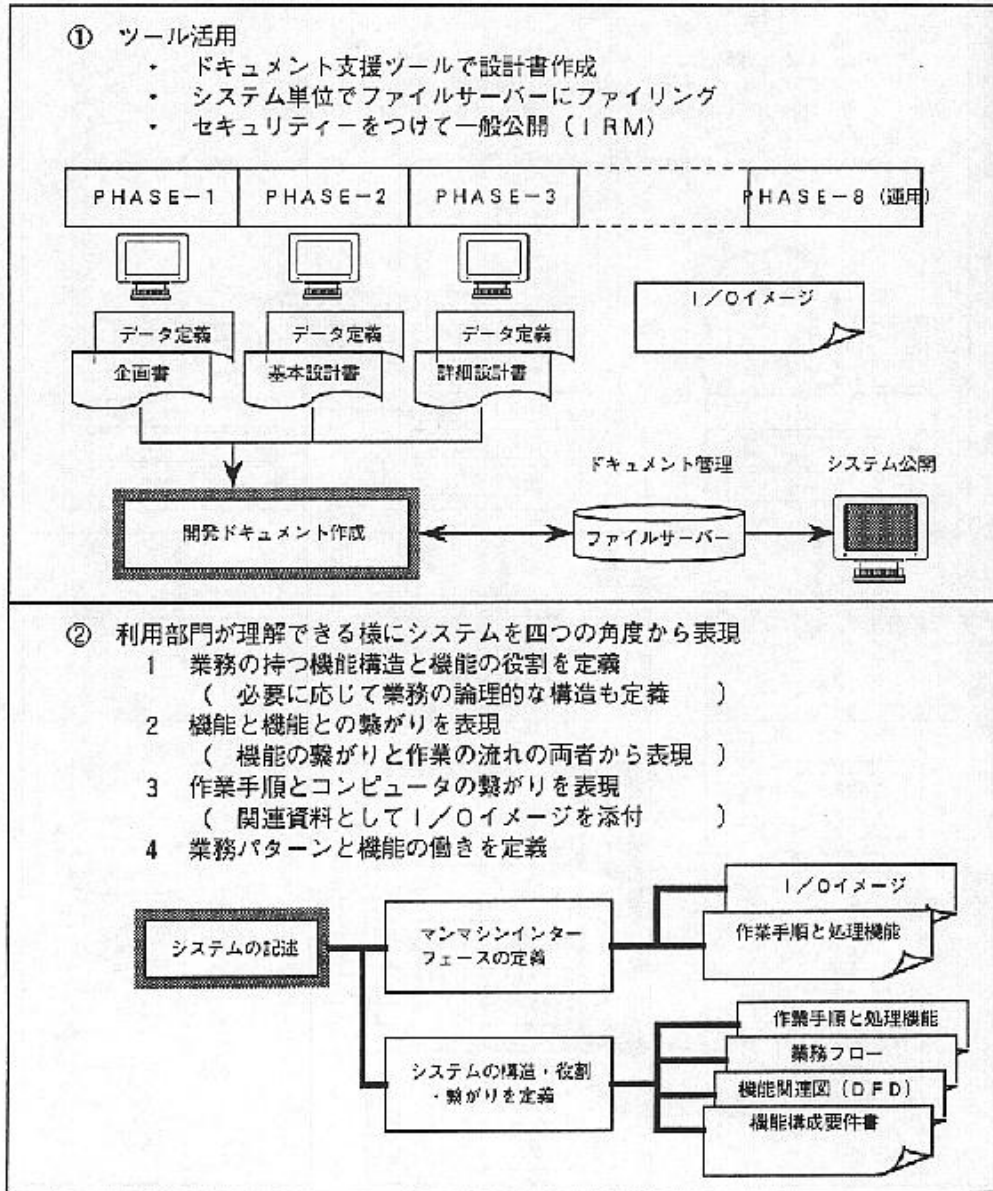


図4-2-2

企業レベルで見たときに、基幹業務のシステム化は一巡しているが、汎用機での集中型システムが大部分であり、システムが大型化し、利用部門にとってはコンピュータ化が業務のブラックボックス化となり、現業部門の改善・改革意識の阻害要因になっているのではないかと思われる。

したがって、利用部門がシステムの仕組みや内容が理解でき現業の改善・改革やシステム運用を行えるようにするためにもドキュメントの公開が必要である。

公開を前提にすると、当然のことながらドキュメントの電子ファイル化が必須となり、そのためにはドキュメント作成ツールの活用が必要である。

また、公開したドキュメントは専門家だけがわかるものでなく、利用部門で理解できるような表現になっていなければならない。

そのために、

- ・ 業務の持つ機能構造と機能の役割を定義
- ・ 機能と機能のつながりを表現
- ・ 作業手順とコンピュータ処理のつながりを表現
- ・ 業務パターンと機能の働き

を定義するという4つの角度からシステムを表現する必要がある。

3) システム設計とはドキュメントを修正すること(図4-2-3参照)

図4-2-3 システム開発標準の考え方
〔設計とはドキュメントを修正すること〕

システムの設計は
「現状の業務をどう変化させるかであり、またその変化を明確にし、
利用部門に正しく理解させ、承認を得ることである。」
また
「よいシステム設計は正しい現状認識から生まれる」
の考え方にに基づき、システム設計の基本を
① 現状分析→ドキュメント化
② システム検討→ドキュメント修正
③ 修正したドキュメントで評価
の手順を踏ませることが必要。

従って、現状調査・分析の段階よりツールにて標準ドキュメントでのまとめ
が必要。
(評価会にて現状と新システムとの対比チェック)

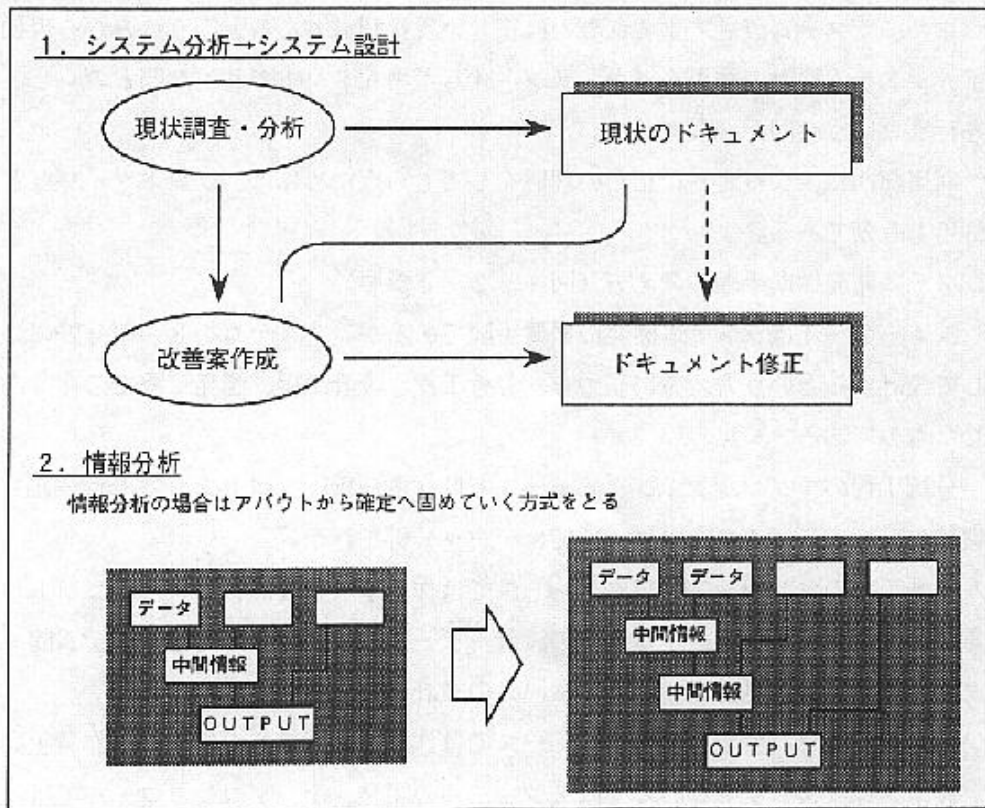


図4-2-3

システム設計は現状の業務を変化させることであり、最大のポイントは利用部門に新しい仕組み、仕事の手順・内容を正しく理解させることである。完成したシステムが利用部門に受け入れられないという悲劇が起きるのは、利用部門が正しく理解した上で、承認していないからである。

また、システム設計で重要なことは正しい現状認識であるということから現状分析、システ

ム検討の結果をドキュメント化しドキュメントで利用部門とのレビューを行うという手順が必要である。

利用部門に、より容易に正しい理解をしてもらうためには、プロトタイプなどの併用も有効である。

4) システム開発標準手順の考え方(図4-2-4参照)

図4-2-4 システム開発標準の手順

	マンマシン インターフェース	プロセス コントロール	データ コントロール
分 析 工 程	<p>作業分析</p> <ul style="list-style-type: none"> ・業務を遂行するのに必要な作業項目と流れの洗い出し ・作業と機能との関連を分析 	<p>機能分析</p> <ul style="list-style-type: none"> ・システムが目撃どおり働くために必要な機能を洗い出す ・各機能の持つ役割の明確化 	<p>データ分析</p> <ul style="list-style-type: none"> ・業務を遂行するのに必要な情報、データの洗い出し ・機能間インターフェースを構成するデータの洗い出し
設 計 工 程	<p>作業手順の設計</p> <ul style="list-style-type: none"> ・効率のよい作業手順の設計と指示メニューの設計 ・指示メニューとシステムモジュールとの接続設計 	<p>プロセスの設計</p> <ul style="list-style-type: none"> ・システムモジュール構造化 ・モジュールの設計 ・プロセスの定義 ・デシジョンテーブルの作成 	<p>インターフェースの設計</p> <ul style="list-style-type: none"> ・システム間インターフェースの設計 ・データ項目の定義、登録更新条件の定義
開 発 工 程	<p>作業手順の登録</p> <ul style="list-style-type: none"> ・個別作業マニュアルの作成 ・メニューテーブルの登録 ・プロセスコントロールのテーブル登録 	<p>モジュールの開発</p> <ul style="list-style-type: none"> ・システムモジュール単位でプログラムを開発 	<p>データベースの開発</p> <ul style="list-style-type: none"> ・インターフェイスFILEの開発 (DBMS) ・更新条件の登録
変 化 へ の 対 応	<p>変更多い</p>	<p>やや多い</p>	<p>少ない</p>
	<p>作業手順の改定</p> <ul style="list-style-type: none"> ・マニュアルの改定 ・メニューテーブルの修正 ・プロセステーブルの修正 	<p>モジュールの差替</p> <ul style="list-style-type: none"> ・改定でシステムモジュールを追加 ・プロセステーブルの修正 	<p>データベースの改定</p> <ul style="list-style-type: none"> ・データベースの改定 ・関連モジュールの改定

図4-2-4

図4-2-4はシステム開発の概要手順であるが、システムを3つの分野に分離して設計するという考え方にに基づき、分析工程、設計工程、開発工程での作業を示したものである。

分析工程のマンマシンインタフェースでは作業分析、プロセスコントロールでは機能の分析、データコントロールではデータ分析を行う。

設計工程のマンマシンインタフェースでは作業手順の設計、プロセスコントロールではモジュールなどのプロセスの設計、データコントロールではシステム間インタフェースという見方でインタフェースの設計を行う。

開発工程のマンマシンインタフェースでは作業マニュアルの作成など作業手順を定義、プロセスコントロールではモジュールの開発、データコントロールではインタフェースファイルとしてのデータベースを開発する。

② オープンシステム開発工程概要

1) オープンシステム開発フェーズ概要(図4-2-5参照)

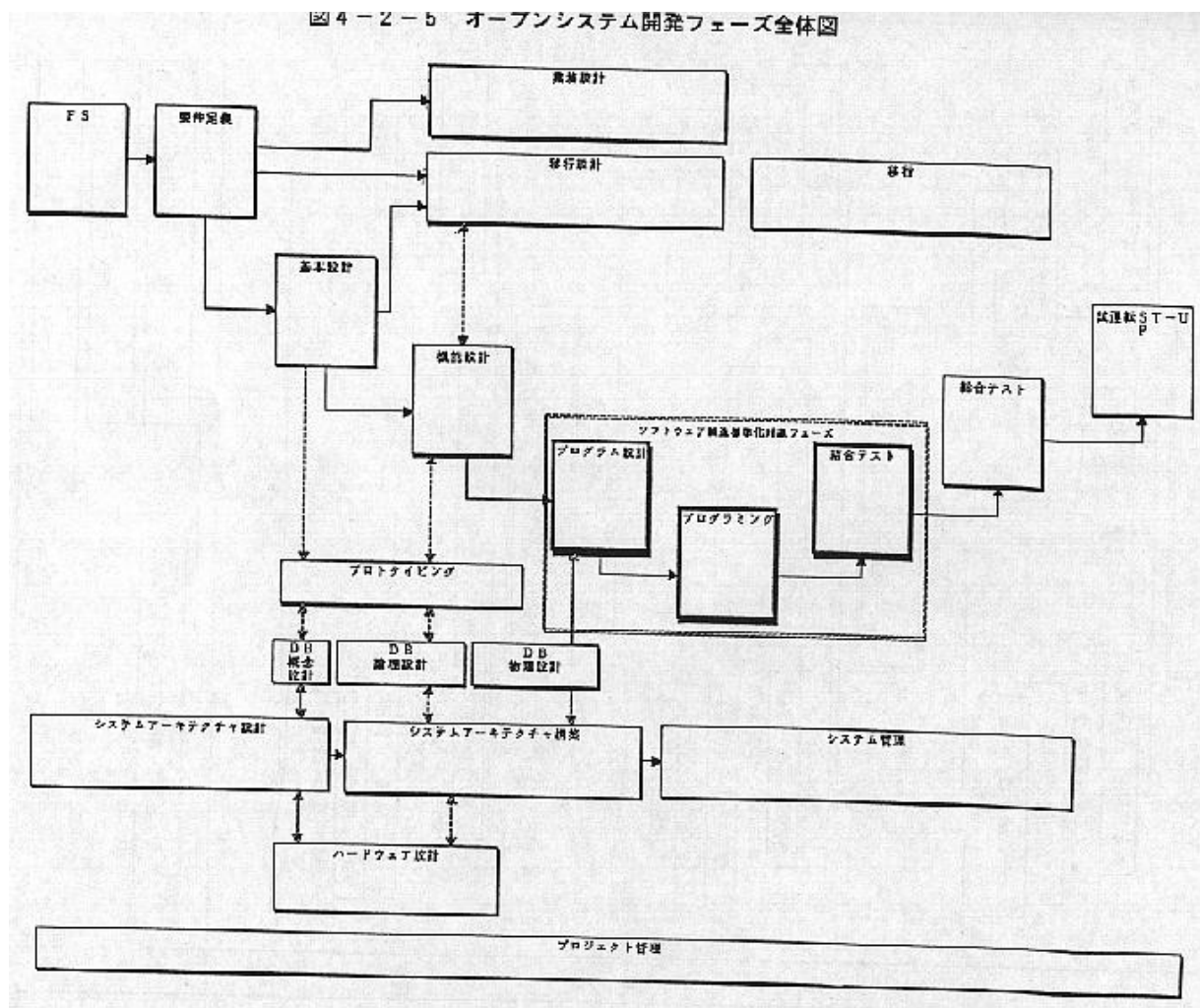


図4-2-5

図4-2-5は開発フェーズ全体図であるが、オープンシステムの特徴としては、プラットフォーム設計と呼ばれることもあるアーキテクチャ設計・構築のフェーズ(図4-2-7)とシステム機器構成などを決めるハードウェア設計(図4-2-8)のフェーズがあることである。

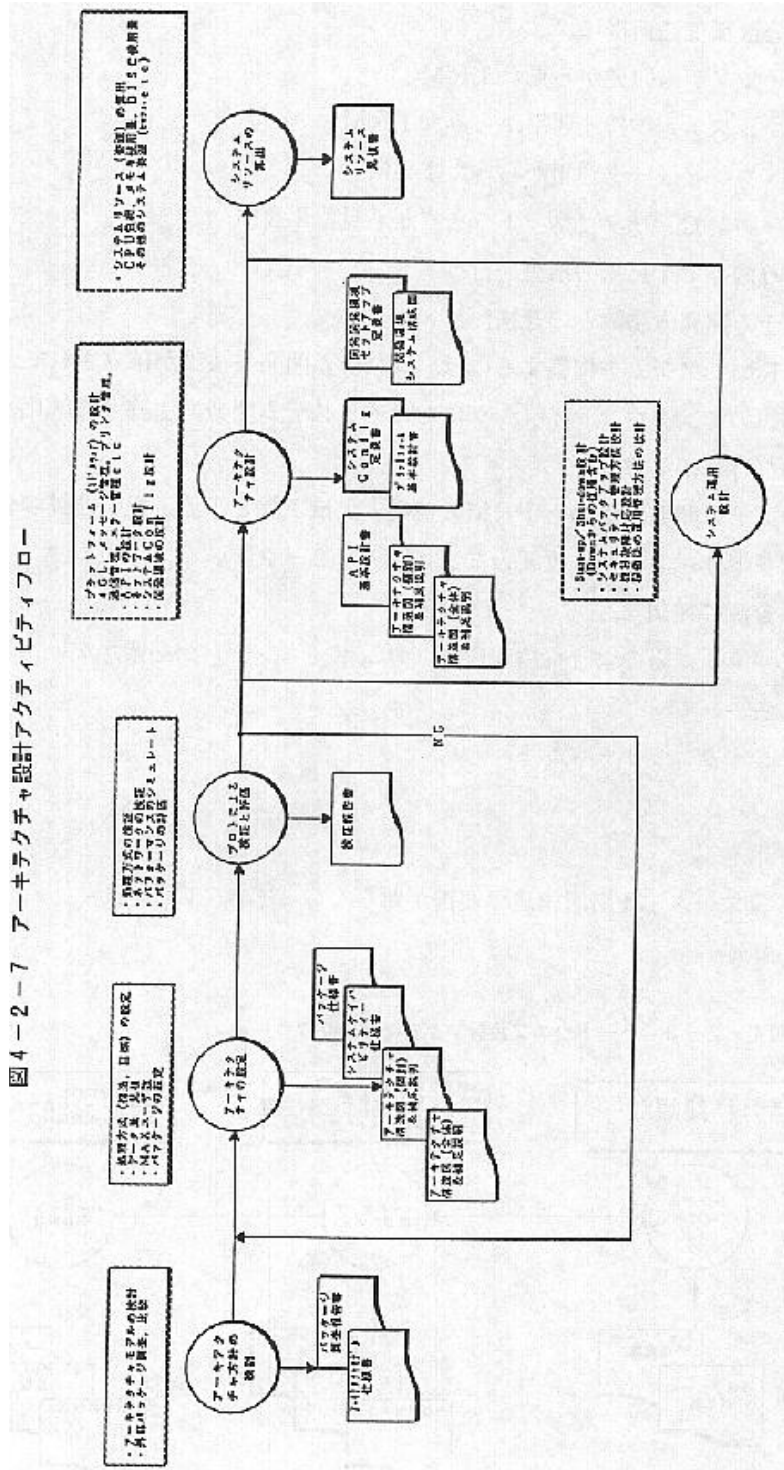


図4-2-7

図4-2-8 ハードウェア設計アクティビティフロー

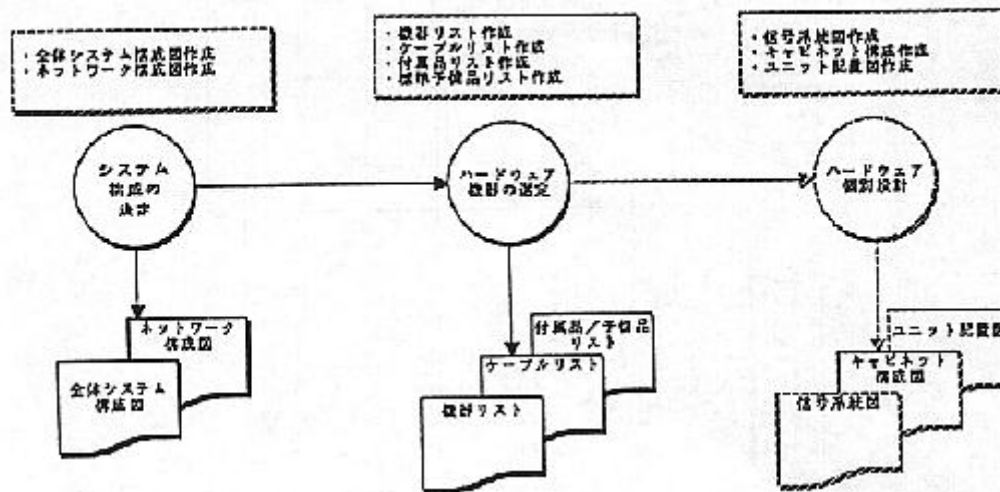


図4-2-8

上流工程のシステム開発ステップ(図4-2-6)は手順としては汎用系の場合とほぼ同じであり、先に述べたシステムを3つの分野に分離するという基本的な考え方が全体を通して貫かれている。

(ファイルが見つからない)

図4-2-6

③ オープン化/ダウンサイジングにおける情報資源管理(IRM)の考え方

オープン化/ダウンサイジングシステムでは利用部門自らシステム開発や維持・運用を行うケースが増えることや分散型のシステムのため維持・管理が複雑になる可能性が高い。

そのためにシステム開発や維持・運用をサポートするツールとして情報資源管理が必須になってくるので、オープン化/ダウンサイジングにおける情報資源管理(IRM)の考え方についても若干言及しておきたい。

1) 基本的な考え方

従来の情報資源管理は、情報システム部門でシステム開発や維持を効率よい生産性を向上させるためのツールという専門的な使い方であった。

オープンシステムでは、エンドユーザーが自分でシステム開発や運用を行うケースやシステム開発に深く関わるが多くなってきている。

したがって、基本機能として次のことが必要になってくる。

a エンドユーザーが情報システムを活用するために必要な情報が検索できる(情報システムの内容がユーザーに理解できる)

- b 情報・データの素性がわかる検索ができる(エンドユーザーのために必須)
- c エンドユーザーが主体的にシステムを活用できる
- d どこに、どのようなシステムを提供しているかがわかる

情報資源管理は「管理のための管理」「情報システム部門だけの管理」は避け、「利用部門サービスの向上」を目的とした考え方で構築しなければならないわけで、次のような観点が必要である。

超分散環境を前提に

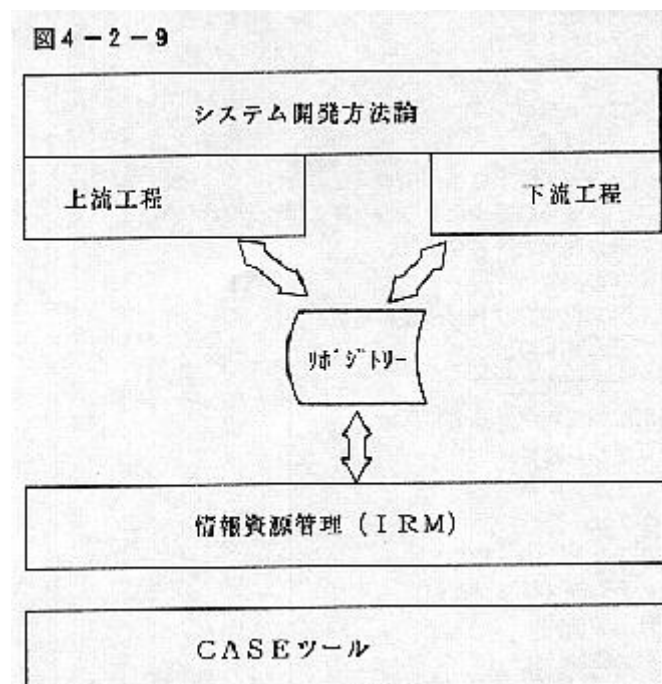
- ・ システム開発から見たIRM
- ・ システム維持・運用からみたIRM
- ・ コンピュータ運用からみたIRM
- ・ コンピュータ、ネットワークから見たIRM

という観点でIRMの構築を行う

2) システム開発方法論との連携

IRMシステムを構築するには、システム開発方法論が確立されていなければならないので、ダウンサイジングにおけるシステム開発方法論(標準化)と整合を考慮しなければならない。

システム開発方法論とIRMの連携、上流工程と下流工程の連携の役目を果たすリポジトリの役割が重要となる(図4-2-9)。



3) 情報資源の種類

まず第一にどのような情報資源(コンポーネント)が必要か洗い出す作業が必要である。

データコンポーネント
ウィンドウ
ボタン
システム:4階層構造の連携が難しくなっている
組織
プロジェクト
ハード仕様:ホスト、ローカル
ロケーション
管理者、所属
プロトコル
使用業務システム
OSのバージョン
DBMSの種類、バージョン
4GLの種類、バージョン
開発ツール
データ提供先のハード(PC……)、ソフト(ロータス、EXCEL……)
担当者
ミドルソフトの種類とバージョン
ネットワーク管理
ローカル側APプログラムのバージョン
システムドキュメント(ユーザーが参照できる)

ただし、なにもかも管理できるわけではないので見極めが必要。

④ 今後の進め方

オープンシステム開発方法論についての概要を述べたが、より具体的なコンセプトや社内標準などを作成するための指針となる基本的な考え方について、体系的にまとめる必要があるので、以下の項目について別途まとめる必要がある。

[オープンシステム(クライアント/サーバーシステム)開発方法論コンセプト編]

1 オープンシステム出現の背景

- 1.1 企業レベルでみた情報システムへの新たな要求
- 1.2 情報システム部門内の問題
- 1.3 オープンシステムの必要性
2. オープンシステムの特徴
 - 2.1 オープンシステムのモデル
 - 2.2 オープンシステムと汎用システム(専用システム)の比較
2. 3オープンシステムの特徴(システムとして何を備えていなければならないか)
処理の分散、管理の集中
3. オープンシステム開発における特徴
 - 3.1 開発方法の考え方(都市計画型)
 - 3.2 開発手順の概要
 - 3.3 利用部門によるシステム運用の実現
4. オープンシステム開発標準化の概念
 - 4.1 オープンシステム開発全体フェーズ
 - 4.2 機能分散型の開発
 - 4.3 システムの3分割
 - ・ マンマシンインタフェース
 - ・ プロセスコントロール
 - ・ データコントロール
 - 4.4 開発ドキュメントは唯一のコミュニケーションツール
 - 4.5 システム3分割基準によるシステム開発標準の手順
 - 4.6 システム開発ステップ
5. 開発ステップの概要
 - 5.1 システム化検討(FS、要件定義)
 - 5.2 システム基本設計
 - 5.3 システム機能設計
 - 5.4 利用部門業務マニュアル作成
 - 5.5 プログラム設計
 - 5.6 プログラム開発
 - 5.7 結合テスト
 - 5.8 総合テスト
 - 5.9 システムテスト(試運転)
 - 5.10 運用・評価
6. データベース設計

6.1 データ分析

データ定義は単に物理ファイルの構成要素を明確にするためだけに行われるのではなく、

エンドユーザーへのデータの素性を公開するためにも必要であるという観点も踏まえて行う

ことがポイントである。

6.2 概念 DB 設計

6.3 論理 DB 設計

6.4 物理 DB 設計

7. システムアーキテクチャ設計・開発

7.1 アーキテクチャ方針の検討

7.2 アーキテクチャの設定

7.3 プロトタイプによる検証

7.4 アーキテクチャ設計

7.5 システムリソース決定

8. ハードウェア設計

8.1 システム構成の決定

8.2 ハードウェア機器の選定

8.3 ハードウェア個別設計

9. プロジェクト管理

9.1 工数見積基準作成

9.2 システム開発体制

9.3 システム開発計画立案

9.4 プロジェクト管理の仕組み検討

9.5 プロジェクト管理サイクル

9.7 プロジェクト評価

10. オープンシステム環境での情報資源管理(IRM)

10.1 情報資源管理の目的

10.2 オープンシステムに必要な情報資源管理の機能

10.3 開発方法論と情報資源管理の関係

10.4 CASE ツールと情報資源管理の関係

10.5 リポジトリの機能

4.3 品質保証

(1) ソフトウェア・システムにおける品質

一般的に言って、ソフトウェア・システムは目に見えにくい“もの”である。そんな事情もあつてか、ソフトウェア・システムの品質については、多くの議論や考え方の提示がなされてきた。

図4-3-1はソフトウェア・システムにおける品質を考えるうえでの、1つのフレームワークを示したものである。また表4-3-1は品質の構成要素を整理して示したものである。

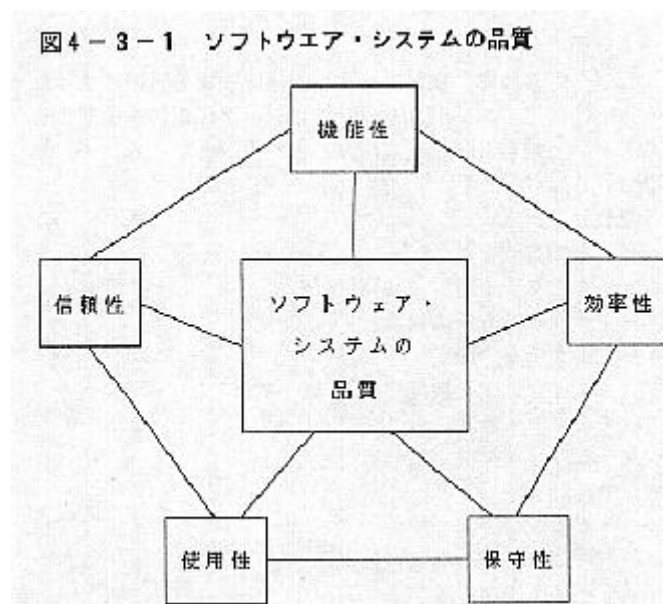


図4-3-1

表4-3-1

表4-3-1 ソフトウェア・システムの品質構成要素

品質の特性視点	構成要素
機能性 (Functionality)	合目的性 (Suitability) 正確性 (Accuracy) 接続性 (Interoperability) 適合性 (Compliance) セキュリティ (Security)
信頼性 (Reliability)	成熟性 (Maturity) 障害許容性 (Fault Tolerance) 回復性 (Recoverability)
使用性 (Usability)	理解性 (Understandability) 習得性 (Learnability) 操作性 (Operability)
効率性 (Efficiency)	実行効率性 (Time Behavior) 資源効率性 (Resource Behavior)
保守性 (Maintainability)	解析性 (Analysability) 変更作業性 (Changeability) 安定性 (Stability) 試験性 (Testability)

実務的には、自らの個別環境を意識した『自社(自組織)の品質標準／基準』をどう検討・設定するかが極めて重要なポイントとなる。

(2)開発方法論(SDLC)の工程モデルにおける品質保証の考え方

ソフトウェア・システム的设计・開発は、何らかの開発方法論に基づく工程モデルをベースに作業が進められているといっても過言ではない。

開発方法論及びその中核として包含されている標準工程(工程モデル)は、各ベンダーごとに若干の相違点(例えば、工程や作業項目の名称の付け方やその内容定義が微妙に異なる等)はあるものの、骨格をなす考え方に大きな違いはない。

図4-3-2は、ウォーターフォール型の工程モデルにおいて、品質保証の考え方がどのように組み込まれているかを示したものである。

図4-3-2 ウォーターフォール型の典型的な工程モデルと品質保証の仕組み

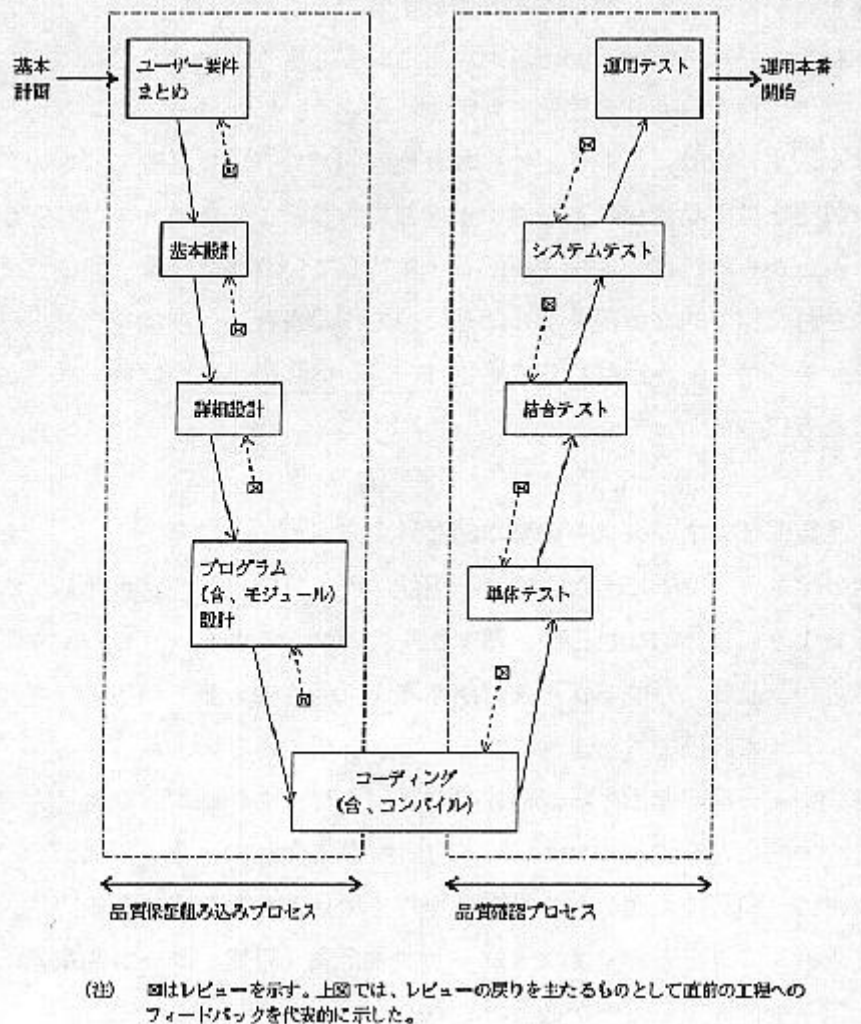


図4-3-2

図で示したように、システム的设计・開発の工程はV字型の構造になっている。前半部分の設計はトップダウンでマクロ(システム)からマイクロ(モジュール)へと順次ブレイクダウンされていく。V字型の底にあたる部分で最少単位の“もの”づくりがなされ、テストはボトムアップでマイクロ(モジュール)から幾つかのレベルを経て、最終的にマクロ(システム)まで実施される。

この大きなフローの中で、前半部分は『品質保証組み込みのプロセス』、後半部分は『品質確認プロセス』ととらえることができる。

また各工程の最後にはレビューが位置づけられ、当該工程の成果物の品質確認がなされることで、全体として手戻りを少なく(理想的には無しに)する小さなフィードバック・パスが用意されている。

全体を貫いている基本的な考え方は以下のとおりである。

すなわち、品質は設計で95%以上規定されてしまうということに対応し、“もの”を作った後で検査し、その善悪を判定するという、いわゆる「事後対応型の品質保証」ではなく、よい設計に基づき定められた仕事のやり方で良品質の“もの”を作るという「未然防止型の品質保証」を強く指向しており、この点に大きな特徴がある。

別の視点からみれば、各企業が所有・維持している当該企業(組織)向けにカスタマイズされた標準的な工程モデルに基づいて作業を行い、環境を作り上げ、遵守することで、そこから生み出される成果物は一定レベルの品質が維持できることになるという考え方につながってくる。

(3) 品質保証マネジメントの基本的な考え方

実際のマネジメントにあたっては、可能な限り品質目標を定量化し、客観的な把握ができるように具体的な目標値、基準値を、システム開発(プロジェクト)の計画時点で設定し、品質に対する保証確認が開発者(受注者)側、利用者(発注者)側双方でビジュアルに評価できるようにすることが肝要である。

蛇足ながら一言申し添える。前述のフレームワーク、基本的な考え方等は、あくまでも一つの考え方を提示したものであり、実際に個別の企業(組織)での適用にあたっては、自らの環境に適合した品質管理対象項目とその測定・評価方法、及び各基準値(目標値)の設定と、マネジメント面でも企業(組織)ニーズや組織風土を踏まえた“一味”を付加する工夫が必要とされる。

(4) 関連トピックス

ここ2～3年、「ISO9000-3」をめぐるホットな議論が続いてきた。この「ISO9000-3」を一言で言えば、「ISO9001」をソフトウェア対象に情報処理業界向けに改定したものと言える。

一般的に製品の品質保証を考える場合には、図4-3-3で示したように対象となる“製品”、製品を作り出す際に適用される“技術・技法”、そしてインフラとしての“品質システム”を三位一体で把握する必要がある。

図4-3-3
品質保証を考えるフレームワーク

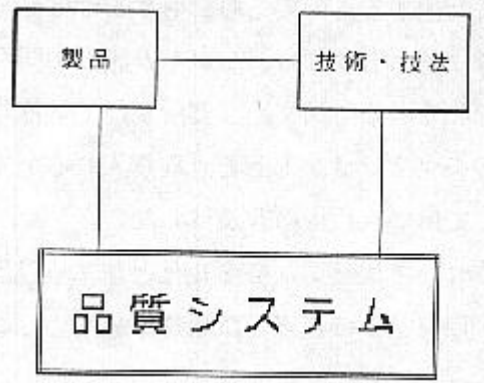


図4-3-3

しかし、「ISO9000-3」では、特定の開発方法論(SDLC)を前提にしているわけではなくと公式表明しているし、周知のようにソフトウェア(システム)の製品としての品質規格を定めたものでもなく、製品を作り出す基盤となる“品質システム”についてのみ規定しているものである。

ところで、「ISO9000-3」については、早ければ1994年末頃までにJIS化されるのではないかと噂を耳にする。日本の情報処理業界はまだ若く、体質も脆弱なので、業界の体質強化のためにはガイドラインではなく規約(JIS化)が望ましいからとの理由説明が流布している。経過はともあれ、この先目が離せないところである。

(5) 品質保証に関する要望

考えるまでもなく、品質保証のテーマは開発者にとっても利用者にとっても永遠の課題の1つである。開発者について言うならば、地道な取り組みを継続することこそが、将来の“優位”を約束し、よい結果をもたらしてくれるはずだ。

利用者について言うならば、システム要件定義として機能はもちろんのこと、操作性や性能を含めた品質目標を明示することで、結果的に速く、かつ安価で“本当に欲しいシステム”を入手することにつながるはずだ。

以上、カスタムメイドのシステムを前提に記述してきたが、多数の顧客を相手に流通しているパッケージソフトについて一言付け加えておきたい。

利用者の視点からの要望は、パッケージソフトの説明書記載項目に品質に関する項目を含めた表示規格が制定され、ソフトの動作及び記述内容についてベンダーが文字通り「品質保証」してくれることである。

これが遵守されれば、利用者は今まで以上により安心してパッケージソフトの購入の意思決定ができる。

『今、一番欲しいのは“ソフトウェア版暮らしの手帖”である』という実感を一日も早く捨てられるようになりたいものだ。

4.4 見積り

(1) 見積りの問題点

見積りは企業(組織)にとってシステム計画や開発の管理上、重要な作業の1つである。見積りは予算獲得はもとより、実際のシステム開発(内作にしる、外部委託するにしる)にあたり、実行計画のベースを定めるものである。

特に平成不況の真只中、情報化投資に対しても経営トップの厳しい目が光る今日において、その重要性についてはあらためて強調するまでもない。

見積りに関する問題点は、結果論から言えば「見積り精度が悪く、なかなか当たらない」という表現に集約されることになるろう。

この主たる原因は2点ある。1つは前提条件としての『対象システム』の要件仕様の確定度合いに起因するものであり、もう1つは見積りに関する技術的なものである。

このうち前者に関わる事項は、システム設計・開発が進行する過程で内容が次第に詳細に確定していく実態を踏まえ、要所要所で見積りを複数回実施すれば見積り精度が向上する事実をマネジメント上どう取り入れるかで解決するしかないであろう。したがって、ここではこれ以上の言及はしない。

一方、後者については適切なモデル化と基準値管理で相当レベルの解決を図ることが可能である。以降、この視点で議論を深めていくこととする。

(2) 見積りをとりまく環境世界

見積りをするうえで考慮すべき対象となる環境世界は一般的に図4-4-1のように構造化し、整理することができる。

図4-4-1 見積りに関連した環境世界の構造

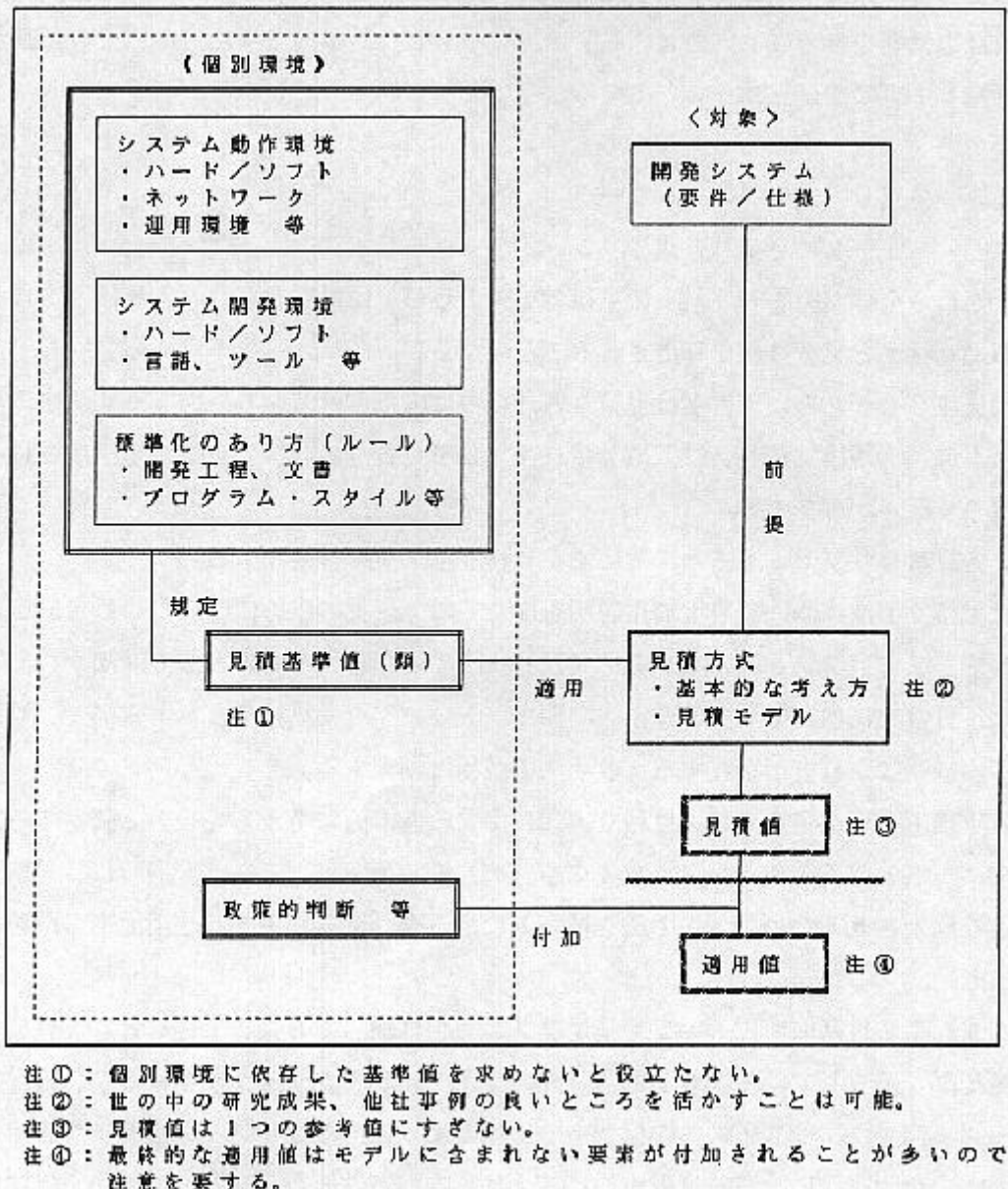


図4-4-1

与件としての開発対象システムに対し、独自の見積り方式を用い、見積り基準値を適用し、見積り値を求めることになる。ここまでが一般的な見積りモデル化可能な世界である。しかし、現実には、この見積り値を重要な参考値としながらも、特に対外的には、各種の政治的、政策的判断が加味され適用値(予算値、契約値、等)として形を変えて実際に用いられる。

見積り値と適用値との間で一線を画したことは、モデルの中にノイズの影響を持ち込みたくないという思いからである。

この図でもっとも重要なポイントは、総論として、仮にこの図のような関係が理解できたとしても、実務的には各企業(組織)の個別環境に依存した基準値を自らの努力で求めないと役に立たないという点である。

もちろん大切な点としては、見積り方式(後述)についても、世の中の研究成果や、他社事例等を参考に、自社(組織)にフィットした基本的な考え方や、それに基づく見積りモデルを確立することである。

(3) 見積り方式のフレームワーク

図4-4-2に見積り方式のフレームワークを示す。

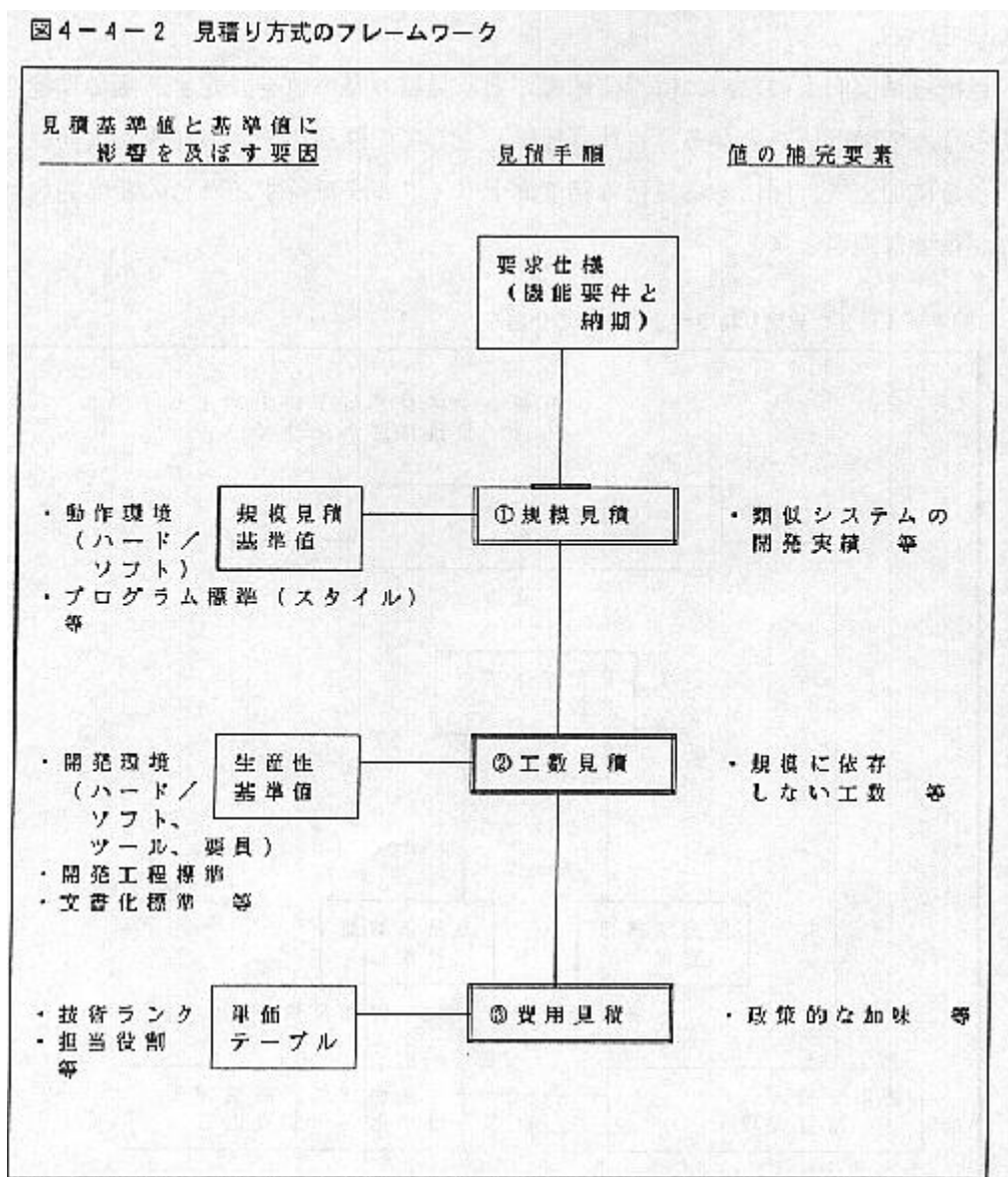


図4-4-2

これら全体が図4-4-1の見積り方式のBOXに対応しており、見積り方式を手順通り行くと見積り値が算出されることになる。

見積りモデルという言葉は見積り方式の中に内包されており、図4-4-2に示すように、規模見積り、工数見積り、費用見積りのサブセットから構成される見積りモデルが存在する。

各見積りサブセットごとに対応する見積り基準値を必要とする。

見積り手法について若干補足説明を加えておく。世の中的には、COCOMOモデルや、ファンクション・ポイント(FP)法などが著名なものとして認知されているし、日常的には類似システムからの類推、トップダウン見積り、ボトムアップ見積り等、様々なやり方が存在する(各手法の説明は割愛する)。

実務上大切な留意点は、見積り方式に関する基本的な考え方を検討・確定する過程でベースとなる手法(カスタマイズを含め)を決めることである。同時に、見積り精度を向上させるために他の手法を補完的に組み合わせて利用する運用面での配慮が望ましい。

また生産性基準値についても補足コメントを付加しておく。工程などについては開発標準が確立している必要がある。そのうえで、ライフサイクル全体とか、設計フェーズ、開発フェーズ、あるいは個々の工程のように、複数レベルの基準値が設定されることが望ましい。これは現実的に言えば、内作/外部委託(どの工程から外注するか)の発注パターンと関連させて)や、業者別の基準値にならないと役に立たない場合が多い。

言わずもがなではあるが、この生産性基準値は単価テーブルとも密接に関連しており、それぞれを独立に考えることは難しい(「この生産性だから、この値段」あるいは逆に「この値段だから、この生産性」といった実態がある)。

単価テーブルについて言えば、技術ランク別に設定されているのが常識的である。

(4) 見積り精度向上のための仕組み／体制

繰り返し強調したように、実際の見積りにおけるキー・ファクタは自社(組織)としての基準値であり、この良悪が見積り精度を左右する。

図4-4-3は 見積り精度向上のための仕組みを示したものである。図中、2つのサイクルが示されているが、内側のサイクルが組織的な基準値管理(PLAN-DO-SEE)を示している。ここでの見積り基準値見直しは1~2年の期間毎と想定している。また外側のサイクルは個々のシステム開発(プロジェクト)における適用を表している。

図4-4-3 見積り精度向上のための仕組み

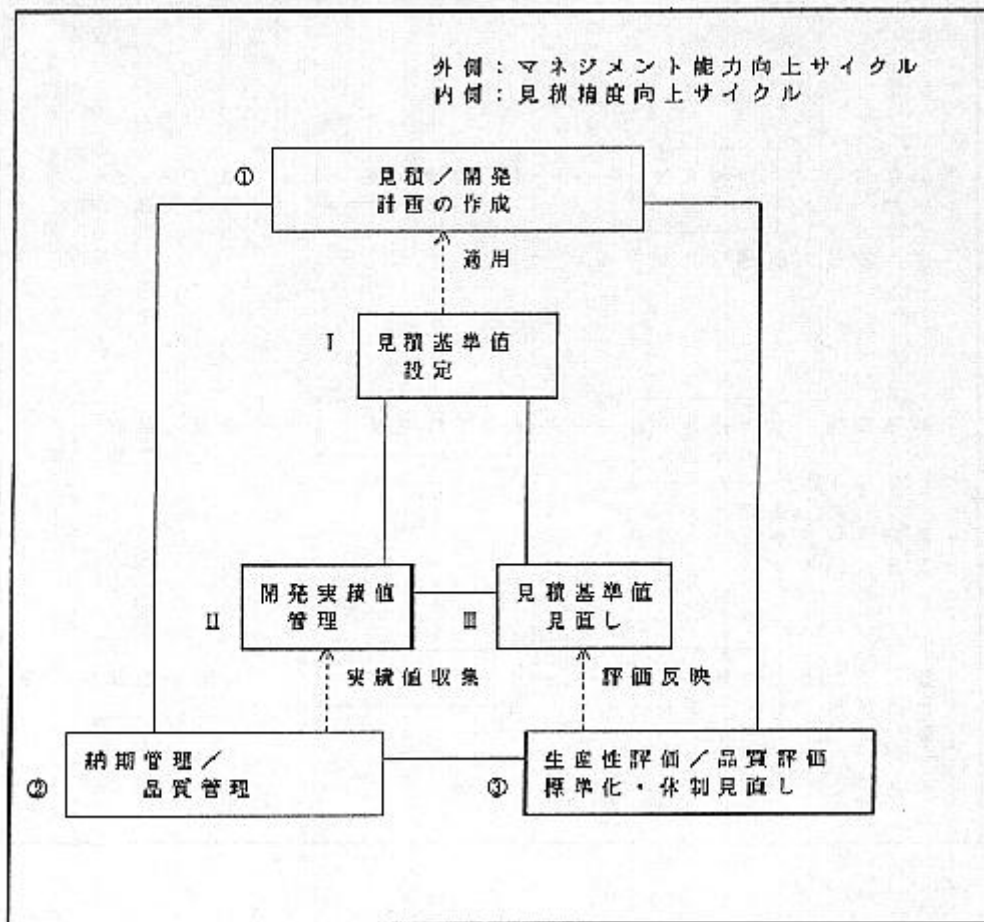


図4-4-3

見積り精度向上のためには、前述のごとく見積り基準値を設定する際の環境を識別するうえでのベースとなる「自社(組織)としての開発標準」の作成が大前提となる。

さらに加えて、図にある見積り精度向上サイクルを維持するための組織的な体制作りが重要な課題となる。

(5) 見積りの“万能薬”はない

「わが社の状況にフィットした見積りの仕組みがどこかに“もの”としてあれば、たとえ高価でも購入したい」という考えを持たれている方はよもやおられないだろうが、心の片隅に淡い期待を捨てきれない人がおられたら、一刻も早く夢から覚めてもらいたいものである。

図4-4-1で示したように、見積りについてその本質を一般的に理解することはできても、こと実務的に言うならば、基準値は個別環境世界を反映したものでなければ役に立たない。この個別環境に適合した値は、当該企業(組織)内で地道に蓄積する以外、管理できないものである。

また生産性指標などについての他社比較は、一般的に言えば、百害あって一利なしである(その理由は、個別環境の違いがわかる場合にはじめて、単なる数値の比較を越えた意味の解釈が可能なのだから)。

夢を捨てきれない人は、“青い鳥”を求めて旅を続けるチルチルとミチルの姿にオーバーラップされよう。

4.5 契約

本項では、特にカスタムソフトウェア(出来合いの市販ソフトウェアではなく、発注者個々の要求を満足するために開発・改造されたソフトウェア)開発維持に関する契約について述べる。

契約に関しては、まだ国際的な標準というものが存在しない。契約締結に至るためには、契約を締結する2者(当然複数の場合もある)間で、契約の内容について合意ができることが必要であるが、ソフトウェアについてはその作業が多岐に渡るということの他に、個々の作業内容の定義がまだ曖昧であり、そのプロセスについて標準化の試みが行われているのが現状である。

そうした標準化活動としては、以下がある。

- ・ ソフトウェア品質:ISO/IEC JTC1/SC7

ソフトウェアライフサイクルプロセス

ソフトウェアプロセスアセスメント

- ・ ソフトウェアの品質保証:ISO 9001 (ISO 9000-3)

- ・ 用語関連(Quality- Vocabulary) :ISO 8404

国内においてもこうした動きに対応し、ソフトウェア開発維持に関する共通の枠組みを規定しようという活動が行われている。例えば、情報処理振興事業協会においては、「共通フレーム検討委員会」が設置され、昨年末に検討結果の報告を行っている。また契約そのものについても、産業構造審議会において検討が行われ、昨年6月、「ソフトウェアの適正な取引を目指して」のレポートが提出されている。またソフト品質についても、1993年11月に「日本品質システム審査登録認定協会」が日本における認定機関として認可されている。

なお、SC7の活動内容は以下のとおりである。括弧内は対応する国内委員会等を示す。

◎ WG2

ソフトウェアドキュメンテーションの検討(ISO6592)。対応する JIS は X0126 であり、「応用システムの文書化要領」の見直し中である(情報処理学会 企画調査会SC/WG2)

◎ WG4

CASEツールの評価選択プロセスについて検討中(情報処理学会企画調査会SC7/WG6、SC7 /WG11)

◎ WG6

ソフトウェア製品の評価とメトリクス(ソフトウェア品質の定義、尺度等のガイドラインの検討)(情報処理学会企画調査会SC7/WG6、日本規格協会情報技術標準化センタ(INSTAC)で品質の定義のJIS化を実施中)

◎ WG7

ライフサイクルプロセスモデルのDIS化作業中(情報処理学会企画調査会SC7/WG7、及び、通産省構造審議会「システム開発共通フレーム」)対応国内委員会等が2組織あり、2元化の恐れがある。

◎ WG8

ライフサイクルプロセス(構成管理、品質保証等)について検討中

◎ WG9

ソフトウェアの分類について検討中。

◎ WG10

ソフトウェアのプロセス評価方法について検討中。カーネギーメロン大学のSEI等の手法をベースとしたアセスメント手法評価基準の検討を実施中(日本規格協会情報技術標準化センタ(INSTAC)プロセス評価標準化委員会)

◎ WG11

CASEツール間のデータ交換、リポジトリデータ蓄積を目的としたソフトウェアエンジニアリング用データ記述について検討中(情報処理学会規格調査会SC7/WG11)

(1) 契約の基本的考え方

我々ユーザー(発注者)にとっては「より良いものを、より安く、より早く」入手することが望みであり、それを保証してくれるのが契約である。そのためには、発注者の意図を正確に契約相手先(受注者)に伝えること、発注金額や作業遂行に必要な期間等を正確に見積ることが必要である。

一方、受注者からみた場合は、受注作業内容が明確で、その作業遂行に必要な稼働が正確に見積れること、曖昧な作業が排除されていることが必要である。また、日本のソフトウェア契約については、建築や製造業関係のものを土台として発展しており、必ずしもソフトウェア開発・維持工程の実態を反映したものとなっていない。契約形態と作業の実態が一致しておらず、過酷な条件で作業を強いられるのも散見される。

このように、契約は、それを締結する当事者双方にとってメリットとなるものであることが望ましい。一方のみが利益を享受するものは好ましくなく、そうした契約を長続きさせることは困難である。また、最近の国際化の潮流及び経営の透明化の観点からすれば、契約の公平性の確保が必要である。すなわち、発注先の選定の方法がオープンであること、発注者と受注者が責任を分担したものであること、等である。

本項では、上記のような背景から、標準化活動そのものではなく、どのようなことを念頭に置いて契約を行えばよいのかという観点から、要求仕様の作成方法、発注金額の見積り方法について述べ、最後に産業構造審議会のレポートについて若干のコメントを述べる。

(2) 要求仕様の作成方法／変更方法

「ソフトウェアは、人間の頭の中にある概念を具現化して処理を行うもの」である。ソフトウェアを開発する人が使う人1人である場合は何ら問題はないが、使う人が別の人であったり、ソフトウェア作成を他者に頼る場合は、いかにその概念を正確に相手に伝えるかが大切であり、下記のようなことに留意が必要である。

① エンドユーザーの要望を正確に仕様に反映

発注責任者と出来上がったソフトを使うユーザー(エンドユーザー)が異なることが、特に大きな企業で見受けられる。こうした場合、往々にしてエンドユーザーの意図が受注者に正確に伝わらないことに起因する下記のような問題が生じる危険がある。

- ・ 曖昧な仕様の解釈がユーザーの意図したものと異なり、納入されたプログラムが使い物にならない
- ・ 開発工程の後になって仕様の変更が必要となり、完成遅延、開発費増大を招く

このような、問題を避けるためには、発注時の仕様の確定なり、開発の過程において、エンドユーザーの参加を求め、仕様変更等について責任をもって対処することが両者の利益にとっても大切である。

② 仕様としての完成度が高いこと

上記に関連するが、受注側からすれば、曖昧な仕様を提示されたのでは、その業務遂行にどれだけの作業がかかるかわからず、かつ作業遂行途中での仕様変更による手戻りの危険が極めて大きいということになる。そうした危険を許容して仕事を受注するためには、最初から(水増しした)大きめの受注金額を提示するなどの自衛策を講じるが必要になる。一方、ユーザーからすれば、大きな金額を請求されているという疑念が生じる。こうした相互不信が今までの契約では根底に屢々あったように思われる。

こうした相互不信を解消するためには、ユーザー側も発注に先立ち明確な仕様を提示することが必要である。自社のみで仕様を確定できない場合は、以下のような手当てを講じるべきであろう。

- ・ 受注側に仕様の確定を依頼する場合は、請負契約とは切り離して、相当の期間SE(システムエンジニアリング)契約で対応してもらおう。
- ・ 受注社とは別の外部の専門家を利用し仕様を確定してもらおう。

また、受注側もプロトタイプなどを通して、製品の完成イメージを事前にユーザーに提示することにより、ユーザーの理解を助け、開発の手戻りを軽減する努力を行うことが重要であろう。

仕様確定に至る過程で参考になるのは、米国政府機関(国防等は除いて)が一般的に利用しているRFP(Request For Proposal)である(図4-5-1参照)。これは広告調達の一環として組み込まれているが、考え方はいろんな場合に適用できる。

図4-5-1 米国のRFP適用例（米国政府の場合）

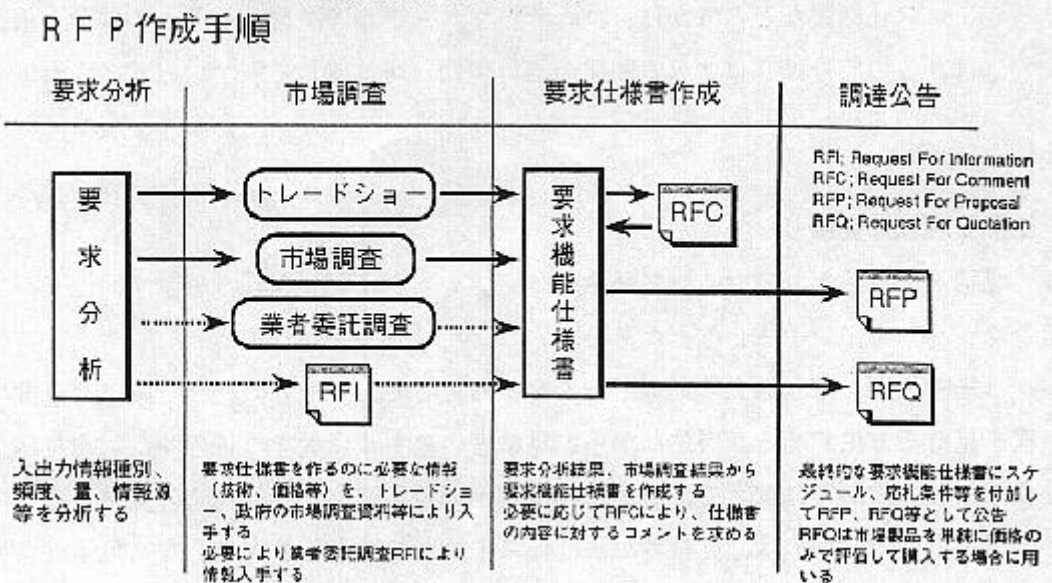


図4-5-1

(3) 契約(開発)金額の見積り方法

ソフトウェアについては伝統的にドンブリ勘定の要素が強く、多少とも科学的手法に基づく見積りが行われるようになってきたのは、ごく最近である。ソフトウェアの自動合成が叫ばれてから久しく、4GLなどかなり高い生産性を期待できるものも環境として出てきているが、作業遂行にどれだけの人手がかかるかが契約金額決定の一番のファクタであることに変わりはない。

① 発注作業の明確化

ソフトウェアの開発保守のライフサイクルは一般的に図4-5-2のように表される。見積りタイミングとその見積りの用途の一般的対応を併せて示す。

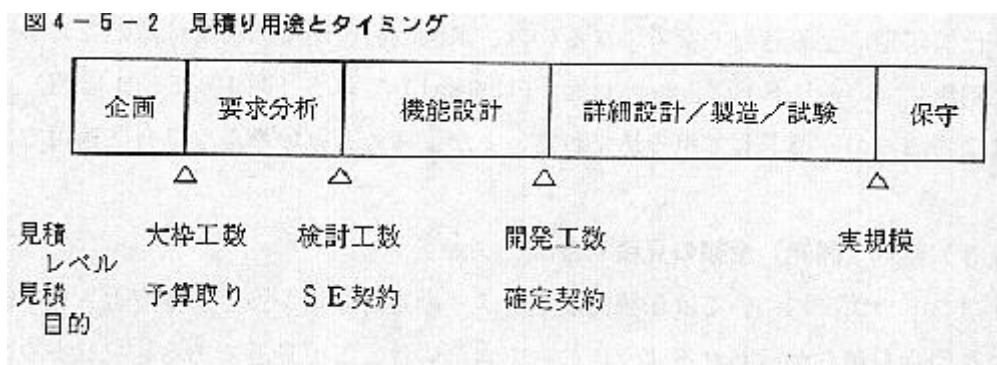


図4-5-2

発注作業の確定のためには、上記のうち、どの部分を発注するのか明確にすることの他に、以下の事項を明確にして提示することが必要である。

- ・ 企画、要求分析については、どちらの責任でどれほどの作業をやるのか(この部分は基本的には発注側の責任部分であるが、システム改善提案などという名目で受注側が代行しているのも事実。経費について意識合わせが必要)
- ・ 外部仕様条件
- ・ 開発すべきプログラムが新規／改造／移植のいずれか。及び、プログラム構成条件
- ・ 各工程でやる作業の内容は何か。特に試験工程では、他社製造の製品との連動動作確認作業などが契約に含まれるのかなどは極めて大きな問題である。大規模なシステム開発においては、こうした連動試験は切り出しSE契約等の別契約とするのが、無用の混乱を回避するうえから好ましい。また、連絡調整ルールや分担 組織構成など開発体制を明示することも必要である。
- ・ 保守契約についてはいつまで保守を行う(依頼する)のか。その期間を明示する
- ・ 生産物は何か。具体的には、請負契約に含めて納入すべきものとして、プログラムファイルの他に何があるかを明らかにすることである(機能設計書、インタ フェース仕様書などのほかに、ユーザーマニュアル等を含むか否かの問題。最近では、この部分はプログラム開発の請負契約と切り離して別契約とするのが増えてきている)

② 開発規模の見積り

見積りの手法としては、「経験法則」「プロトタイプング」「分割予測法」「ファンクションポイント法」などがある。

「経験法則」は、過去の類似データを参考として、見積者の経験と勘により開発規模を見積る方法である。当然ながら、見積者のスキルに大きく依存する。本見積値は企画などの段階では重宝であるが、実規模との差を抑えたい契約時の見積りとしてそのまま単独 で使うのは、見積者がよほどスキルがないと差が大きくなる危険性がある。他の方法との併用を心がけるべきであろう。

「プロトタイプング」による見積りは、製品の出来上がりのイメージを見るため一部の機能を試験的に 実現する際、そのデータを基に全体の規模を見積ろうというものである。当然、障害処理などを含めて実質処理を大きく省略しているので、適用としては企画段階など開発工程の初期に利用するのに適している。

「分割予測法」は、機能／モジュール等の分割を詳細に行い、その後、代表的なモジュールのいくつか の部分を実際に開発してそれを積み上げ、全体の規模を予測する方法であ

る。実際に開発する代わりに過去の類似のモジュールの実規模との比較を行い予測することも行われている(むしろこれが一般的)。

「ファンクションポイント(FP)法」は、プログラムとして実現する機能の難易度に応じて点数をつけ、その点数に応じて経験則的に規模に換算しようというものであり、ソフトウェアの複雑さ、機能の多さによりソフトウェアの規模を見積ろうというものである。このため、定式化された手法である、開発手法に依存しない(要求仕様から求めることができるため)、開発の初期から適用できる、などの特徴を有する。

FP法は、A. J. Albrecht が考案したものであり、IEEE のチュートリアル教材として活用されてきた。FP法としては、IFPUG法、SPR法、JISA法、IPA法、フィーチャポイント法、MarkII FP法などがある。こうした様々な方法を統一化していこうという動きも活発化しており、米国の IFPUG (International Function Point User Group、約 470 会員)や欧州のEFPU G (European Function Point User Group) が活動を行っている。日本では、JISA、IPA及び COSDESが適用可能性の検討を行っている。

FP法の適用分野としては、ビジネス分野のアプリケーションプログラムが主流であり、多くの実績をあげている。ただ、OSや通信プログラム(通信処理プログラム)などの分野への適用では、見積り精度の面で改良が必要との声もある。

[参考]

・ファンクションポイント(FP)法の算定手順

手順1:機能ごとのデータ抽出

要求仕様書や機能仕様書からソフトウェアシステムを以下の5つの機能に分解して、それぞれの関係するデータを抽出する。

- ・ 入力
- ・ 出力
- ・ 外部からの問い合わせに対する応答
- ・ 外部データベースへの参照
- ・ (自ソフトウェアシステム内)自己データベース維持

手順2:データに点数付け

データごとにあらかじめ決められた機能の豊富さに対応する点数をつけ、総計(UFP)をとる。

データに与えられる数値をファンクションポイント(FP)と呼ぶ。

手順3:システム特性の評価

システムの重要度、複雑さを大きく左右する、オンライン処理、分散処理、性能条件等、14項目のシステム特性を5段階で評価し、その合計(AVF)をとる。

手順4:総合点算出

次式により、最終的なファンクションポイント(FFP)を算出する。

$$FFP = UFP \times (0.65 + 0.01 \times AVF)$$

(4)業者選定

「より良いものを、より安く、より早く入手」するためには、信頼のおける業者をどう選ぶかがポイントになる。特に、独自仕様に基づいたカスタムソフトウェアを発注しようとして信頼できる業者の心当たりがない場合、その問題は深刻である。ソフトウェアの場合 はハードウェアと異なり、過去の実績がなくても、しっかりした製造ラインをもたなくても、極言すればどんなところでも仕事ができるという面をもっており、どのような基準に基づいて業者の選定を行うかがポイントになる。

選定のポイントは、「しっかりした開発技術を有する会社は良い製品を作成する」ということであり、開発方法、試験方法、ドキュメント作成方法などに、定式化された、系統だった開発技術、管理方法を採用しているかにある。

よいソフトウェアを開発する技術を向上させる(したがって技術を計る)方法として注目されているのが、ソフトウェアプロセスアセスメント(SPA)である。SPAの代表的なものとして、米国カーネギーメロン大学ソフトウェア工学研究所(CMU-SEI)の アセスメント手法がある。これは、ある評価項目に基づいて、どれほどユーザーニーズを効果的にプログラムに変換する技術をそのプロジェクトが有するかを計り、その結果に基づいて、さらにプロジェクトの能力を高めるための施策を打つというものである。評価の観点(したがって高めるべき技術)は、新技術/ツール、資源、標準化、品質保証、構成管理、見積り、スケジュール管理、追跡/計測、組織化、教育など多岐にわたり、現在85の項目により5段階の能力レベルつけを行う。SEI-SPAの評価はかなり厳格であり、能力判定を受験した企業の90%以上が一番初期的段階の(渾沌的マネジメント)レベルに位置づけられている。本方式は米国政府一部機関(DoD等)でも活用されており、応札希望業者はSEI-SPA受験が必要となっている。

(5)産業構造審議会のレポートに対する見解及び今後の対応

通産省産業構造審議会情報産業部会では、昨年6月に「ソフトウェアの適正な取引を目指して」と題して報告書を発表し、契約問題と価格問題について、現状の問題を分析し今後のソフトウェアのあるべき方向の提案を行っている。当委員会においても、ソフトウェア開発にかかわる問題の重要性を鑑み、数度にわたり委員間で意見を交換するとともに、オープンフォーラムにおいて当該報告書の概要と当委員会の考え方を発表し、会員の意見をうかがった。以下、その中での意見を述べる。

【委員会での意見】

(1) 報告の主旨(特に今後目指すべき方向)には、本項でこれまで述べてきた方向とも一致しており、大枠で賛成である。ただ、背景認識、詳細な記述レベルで補足なり修正も必要である。

(2) 機能ベースの市場価格決定の考え方は本来の方向ではあるが、パッケージソフトが普及しておらず、ベンダーとユーザーの作業分担が曖昧な現状では理想すぎる。まず、パッケージソフトを普及させる方策などが大切であるが、そのためには安心してパッケージソフトを利用できる環境を整備することが必要である。

パッケージソフト普及のためには、ベンダーの一存による互換性のないバージョンアップ等は慎む必要がある。ユーザー協会としてもそうした問題情報を積極的に収集公表することがユーザー保護の上から大切である。またユーザー自身もカスタムソフト利用からパッケージソフト利用への方向転換を急ぐべきである。

(3) ユーザーがパッケージソフトに対して持つ疑念は、「品質の保証がない」「いつまで使えるかわからない(いつ突然供給停止になるか不安)」「開封契約が一方的すぎる(動くかどうかは開封しないとわからない)」などにある。この不安が解消されない限りは全幅の信頼をパッケージソフトにおくことはできない。

(4) 本報告の内容は、ユーザー側にもそれなりの技術的・組織的体力があつて初めて達成されるものが多い。中小企業では主旨には賛成であるが、そこまではなかなか対応できないのが実情である。本報告をもって硬直した方向に行くことなく、柔軟な対策を打ってほしい。

(5) コストプラス法ではソフトウェアの価値や品質・生産性の要素を価格に反映することは困難といっているが、いろんな対策はある。あまりに市場価格誘導を追い詰めると、ユーザーが欲している技術/機能でなく、いわば商売優先のマーケット論になる危険がある。

【フォーラムでの意見】

(1) メインフレーム系のソフトウェア開発を前提にしたレポートのように思える。すなわち、ユーザー側にしっかりした技術力があつて初めて対応できることである。中小の企業にとっては、何をやりたいか明確なものがない場合や、あつてもそれをうまく表現できないケースが多

い。その場合にはプロトタイプによる方法等が有効であるが、いずれにしても手戻りがある。そうした状況を反映したものになっていることが必要。

(2) 種々の業界のソフトウェア開発については、その業界独特の慣例や用語などがある。用語の定義などをこうしたユーザーに強いるだけでは問題は解決しない。

(3) 仕様確定へのエンドユーザーの参加は、発注側にとっても受注側にとっても有意義なことである。ただし、それが強制となつてはいけない。

終わりに

従来から標準化活動についてはいくつかの側面から論じられてきた。ソフトウェアの標準化活動に批判的な代表例を上げると、以下ようになる。

- 1) 標準化活動は技術発展の足を引っ張るもの(時間がかかるし、標準ができるころは時代遅れの技術となるという意見)
- 2) 標準化活動は特定のベンダーの製品を普及させるためのもの(帝国主義的に時代遅れとなった製品を技術が遅れている国・地域へ押しつけるものであるという意見)
- 3) 役に立たないもの(標準のものを使ってもそれだけではシステムが構築できない、標準と言いながらベンダーにより微妙に異なる、本当に必要な部分の標準化がなされないであまり関係のない分野の標準化に力が注がれている、等という意見)

こうした意見は、一見、的を射たものにも見えるし、ユーザー側に立ったもののようにも見える。しかし、よく考えてみると、ベンダーがユーザー囲い込みのために故意に強調宣伝したものであったり、ユーザー自身の無知による曲解も多い。例えば、オープン化と言われても(実際は相当コスト高になっている可能性もあるにもかかわらず、1ベンダーで済んでいるので)必要がないと考えているユーザーも案外多いのではな いかと思われる。

本部会では、標準化そのものの活動状況の調査ということではなく、標準化活動自体の意義、ユーザー側から見た問題点(標準化活動自体に問題はないのか、抜けている事項はないか、等)、標準に準拠したものを採用したときの問題点(コスト高とならないか、性能品質上問題ないか、複雑すぎないか、等)などに関して、われわれ部会委員自身の問題認識を深めながら検討を行ってきた。

本報告書を作成するに際しては、問題の所在を明らかにすること以外に、ユーザーから見て本来あるべき姿(ベンダー側の対応のあり方、ユーザーの対応のあり方双方)、ユーザーの社内標準作成等にあたって留意すべき事項等についても触れるように配慮したつもりである。個々の標準化項目ごとに置かれている状況が異なるため、記述内容が異なったり記述の詳細さのレベルが合わないことや重複が多いことなど、読み難い箇所も多々あるとは思いますが、ご容赦願いたい。本報告書が、日本情報システム・ユーザー協会会員をはじめとしたユーザーの方々の製品調達やシステム化等の一助となれば、われわれ部会委員の望外の喜びである。