# 情報処理に関する標準化調査報告書

昭 和 57 年 3 月

JIPDEC

財団法人　日本情報処理開発協会

この資料は，日本自転車振興会から競輪収益の一部である機械工業振興資金の補助を受け，「昭和５６年度情報化の推進に関する調査研究」の一環としてとりまとめたものであります。

# はじめに

近年わが国におけるコンピュータ・システム技術，特に分散処理，日本語情報処理，データベース，マイクロコンピュータ等応用技術の進展が著しく，その利用においても，産業分野における生産事務のオートメーションを始め，社会や国民生活における公共情報処理システムの形式へと拡がりを見せている。

このような情勢の中で，情報処理分野の標準化がますます重要となっており利用の進展に応じ，新規項目の追加，事務分野との対応等が必要であり，特に非関税障壁問題ともからんで，ＩＳＯ等国際規格に対しての整合性や積極的な参加が要請されるなど，従来の進め方等について再検討の時期にきている。

このため，今後の対策の検討に資する目的で，当協会「情報処理標準化調査委員会」では，ソフトウェア関係の標準化について，ソフトウェアの分類と体系，ソフトウェアの効率的な生産・利用のための標準化項目等について調査検討を行った。本書は，この調査結果をとりまとめたものである。

最後に本調査にご協力いただいた関係各位に深く感謝し，厚く御礼申し上げる次第である。

昭和５７年３月

## 情報処理標準化調査委員会

| | 氏　名 | 勤務先及び所属・役職 |
|---|---|---|
| 委員長 | 菅　忠義 | 学習院大学理学部　講師 |
| 委員 | 川野幸雄 | (社)日本事務機械工業会<br>東芝(株)品質保証部グループ課長 |
| 〃 | 岸川良一 | (社)日本電子工業振興協会<br>富士通(株)ソフトウェア事業部検査部長 |
| 〃 | 島谷和典 | (社)日本電子工業振興協会<br>日本電気(株)ミニコンシステム事業部計画部長 |
| 〃 | 妹尾　稔 | 三井精報開発(株)ＭＢＫ事業本部ＴＳ推進部次長 |
| 〃 | 中村幸雄 | (社)日本ドクメンテーション協会会長<br>インフォーコム技術事務所所長 |
| 〃 | 西木俊彦 | ＦＤＰユーザー団体連合会<br>新日本製鉄(株)情報システム部副部長 |
| 〃 | 保科好信 | (社)日本データプロセシング協会<br>日産自動車(株)横浜工場電算課長 |
| 〃 | 松原友夫 | (社)ソフトウェア産業振興協会<br>日立ソフトウェアエンジニアリング企画室長 |
| 〃 | 溝部悦夫 | 日本電信電話公社技術局データ宅内部門調査役 |
| 〃 | 林　明夫 | 通商産業省機械情報産業局情報処理振興課業務班長 |
| 〃 | 伊東　厚 | 通商産業省工業技術院標準部電気規格課 |
| 〃 | 山村賛平 | (財)日本情報処理開発協会常務理事 |
| 〃 | 山本順一 | (財)日本情報処理開発協会前技術調査部部長 |
| 〃 | 市川　隆 | 同　上　技術調査部次長 |
| 〃 | 難波正之 | 同　上　技術調査部技術課課長 |
| | 事務局 | 同　上　技術調査部技術課 |

# 目　　次


Ⅰ　総　　論 …… 1

Ⅱ　各　　論 …… 1

2.1　ソフトウェアとは …… 3

2.1.1　ソフトウェア分類の考え方 …… 9

2.1.2　「プログラム高度化計画」におけるソフトウェアとは …… 17

(1)　プログラム開発に関する新高度化計画 …… 17

2.2　ソフトウェアの標準化 …… 27

2.2.1　標準化の目的とその考え方 …… 27

2.2.2　標準化項目の体系 …… 28

2.2.3　ソフトウェアに関する標準化項目 …… 35

2.2.3.1　ＩＳＯ，JISCにおける標準化の現状 …… 35

(1)　ＩＳＯの現状 …… 35

(2)　JISCの現状 …… 42

(3)　利用者・生産者間インタフェースにおいて必要な標準化項目 …… 45

A．汎用ソフトウェア …… 45

B．受注ソフトウェア …… 54

C．用　語 …… 65

(4)　効率よく生産・利用するには …… 66

A．設計面 …… 66

B．製造面 …… 75

C．ドキュメントの整備 …… 84

D．ＴＣ４６側の援助 …… 93

E．導入面 …… 93

F．利用者側の管理・運用面 …… 98

G. 保守面での標準化（含管理面）……106
H. 拡張面での標準化項目 ……110
I. 契約面 ……112
Ⅲ ソフトウェアに関連するその他の項目 ……117
Ⅳ ISO／TC97／SC7 の幹事国の問題 ……125
Ⅴ 参考資料 プログラム開発に関する新高度化計画詳細説明資料 ……126

# I　総　　論

## Ⅰ 総　論

我が国においては，1970年代に入る頃から情報処理産業が急速に発展し，1975年頃以後，特にソフトウェアが商品として流通するようになり，1980年代を迎えてますますその傾向が強まりつつある。このような時機に当り，ソフトウェア産業において，いかなる事項を標準化すべきかを組織的に考察し，今後の我が国の標準化に関する施策に積極的に反映させることは，我が国の情報処理産業の健全な発展のために極めて重要なことである。

また，GATT承認という国際情勢の下では，ソフトウェアに関する標準化項目を考察し，国際的に必要と考えられる項目については，ISO／TC97へこれらの項目をnew work itemとして提案し，かつ我が国が主導的にその規格作成に寄与していくようにすべきであろう。

上述のような立場に立って，当委員会においては，ソフトウェアに関する標準化項目について多方面にわたって討議を行った。その内容を以下のようにまとめた。

## Ⅱ 各　論

### 2.1 ソフトウェアとは

#### 2.1.1 ソフトウェア分類の考え方

#### 2.1.2 プログラム開発に関する新高度化計画におけるソフトウェアの分類体系

### 2.2 ソフトウェアの標準化

#### 2.2.1 標準化の目的とその考え方

#### 2.2.2 標準化項目の体系

#### 2.2.3 ソフトウェアに関する標準化項目

##### 2.2.3.1 ISO，JISCにおける標準化の現状

##### 2.2.3.2 利用者・生産者間インターフェースにおいて必要な標準化項目

##### 2.2.3.3 効率良く生産・利用するために必要な標準化項目

Ⅲ　ソフトウェアに関連して標準化が望まれるその他の項目

Ⅳ　ISO／TC97／SC7 の幹事国の問題

Ⅱにおける各節の概要は以下のようである。

2.1で，ここで考察の対象とするソフトウェアとは何であるかを明確にし，2.1.1.において，ここで扱うソフトウェアについての分類体系を如何に考えるべきかの一般論及びそれに関連する種々の問題を述べる。2.1.2ではプログラム開発に関する新高度化計画において，通商産業省，メーカ，ユーザの委員が約10ケ月にわたって議論して得た一つの具体的分類体系を示す。

2.2において，2.1で限定した，ソフトウェアに関する標準化項目を論ずるが，先ず2.2.1では何を目的として標準化を考えるのか，その根本的な考え方を述べ，2.2.1においてその立場から考察するとき，標準化すべき項目を，どのような場において，どのような面から考えるべきかを論ずる。2.2.3では，最初に2.2.3.1でISO及びJISCにおいてソフトウェアに関してどのような項目がどのように標準化されているか，また標準化されようとしているか，その現状を述べる。次に2.2.3.2及び2.2.3.3で，2.2.2の視座から，具体的に必要となる標準化項目を導出し，必要に応じてそれに関して考慮すべき問題を述べる。

Ⅲにおいては，2.1及び2.2において考察の対象から除外したもので，ソフトウェアに関連して問題となる項目を示す。

Ⅳにおいては，1981年12月に開かれたISO／TC97の総会において，SC7 の幹事国スエーデンが，財政上の理由で幹事国を他国に代って欲しいと申し出たことに対する当委員会の意見を述べる。これは，今後のソフトウェアの標準化に対して重大な影響をもつ問題である。

# Ⅱ　各　　論

## 2.1　ソフトウェアとは

コンピュータの誕生と共に生れたソフトウェアは，第2世代までは開発すべきソフトウェアの量はさほど多くなく，従来のパンチカードマシンの配線盤の配線を組むことの代り位としか考えられていなかった。

これの開発が，メーカ，及びユーザに於て問題になり始めたのは，およそ15年前のIBM360で代表される第3世代コンピュータの出現以降である。

しかし，その頃のソフトウェア需要は，主としてメーカがハードと共に提供するオペレーティング・システム（OS）と，顧客毎に開発するアプリケーション・プログラムであり，流通商品としてのソフトウェアの成立はIBMのソフトウェア価格分離政策の実施後まで待たねばならなかった。日本に於て，ソフトウェアパッケージの流通が本格化したのは，ここ2～3年である。

このように，コンピュータの付属物として生れたソフトウェアは，それまで存在しなかった全く新しいものとして捉えるべき特質を持っていたにも拘らず，扱う人のそれまでの経験や立場によって，様々な捉え方がなされてきた。

(1)　ソフトウェアの様々な捉え方

(a)　ハードウェアの対照語として

この言葉の語源であろう。この語感からくる解釈は，明らかに「ハードウェア以外のすべて」であって，漠然とプログラム，ドキュメント類，ノウハウ，サービス，その他ハードウェア以外のあらゆるものを含む。このような広義の捉え方は，法的保護の項で後述するように，プログラムとほぼ同義とする狭義の捉え方と並んで，代表的な捉え方である。

(b)　利用技術という捉え方

ソフトウェア出現の初期に，それが利用技術と訳されたところから，ソフトウェアはコンピュータを利用する際に必要な操作ノウハウと捉えられ

たことがあった。この言葉は色々な意味で誤解を招き易いので，最近はあまり使われていない。

(c) 生産者としての捉え方

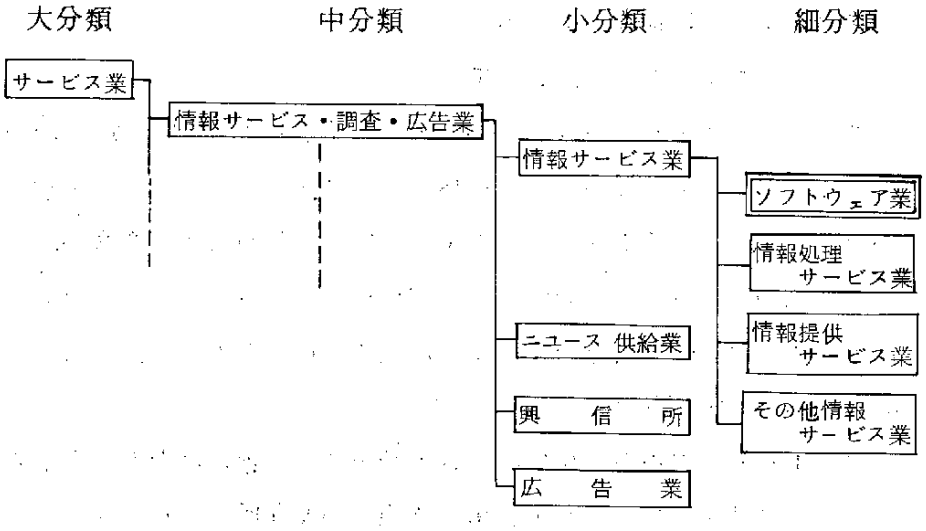
ハードウェア・メーカが，オペレーティング・システムのようなコンピュータの機能の重要な部分を受持つソフトウェアを生産し，供給する場合は，ハードウェア・メーカの伝統と，ソフトウェアを含めたシステム全体の機能，品質，性能に対する顧客の期待から，当然ソフトウェアを工業製品の一つと考える。

このような立場から見れば，ソフトウェアはエンジニアリングの対象であり，現実にソフトウェア・エンジニアリングによって，それは技術的に大きな進歩をもたらした。

しかし，後で述べるように，法律上，あるいは産業分類上などの別な見方からは，それは明らかに「工業製品」としての位置づけがなされていない。

(d) 産業分類上の捉え方

行政管理庁がまとめている日本標準産業分類では，ソフトウェアは，次ようにサービス業の一部として位置づけている。



また，小分類に於て情報サービス業を，ニュース供給業，興信所，及び広告業と同例に扱っていること，情報サービス業の中に，処理手段（ソフトウェア）と成果物（処理サービス，提供サービス）をそれぞれ業とするものが同居していること，などは興味深い。

「ソフトウェアとは何か」は今日でも屢々論じられ，その際の論点の一つは「インタンジブルであるか否か」であるが，この産業分類では，ソフトウェアをインタンジブルな側面から捉えていると考えられる。

ソフトウェア産業発生の初期に於ては，ソフトウェア業の実態はプログラマ派遣業であったから，当時としてはこの捉え方は己むを得なかったであろうが，請負契約が増え，流通商品を持つに至った今日では，やゝ不適切な感を免れない。

(e) 成果物の法的保護からみた捉え方

現在最も論争の活潑な問題の一つであり，その調査研究の成果はソフトウェア産業振興協会のソフトウェア法的保護調査委員会の成果に俟つところが大きい。その中間報告書の第Ⅱ章には，法律や著名な著作物上に記述されているソフトウェアの定義8件が抄録されており，これらを大きく次の2つに区分している。

①ソフトウェアを，プログラムとほぼ同一視する考え方（狭義のソフトウェア）

②ソフトウェアを，プログラムを含むコンピュータの効果的使用に必要な知識・技術と見る考え方（広義のソフトウェア）

そして，ソフトウェアの法的保護の限界を論ずる場合には，その保護が及ぶソフトウェアの外延上の広がりのボーダーラインを明確にしておく理論上の必要があり，この観点から，ソフトウェアの範囲は広義に定義されるのが妥当である，と報告書は述べている。(1)

注(1) (社)ソフトウェア産業振興協会　ソフトウェア法的保護調査研究特別委員会ソフトウェア法的保護の調査研究に関する中間報告　昭和56年3月（P11～15）

さらに，同報告書は，法的保護の対象としてのソフトウェアが持つ法的特質について，次の2つを挙げている。

(i) 知的無体物としての性格

これに対する権利として無体財産権があり，それはさらに工業所有権と著作権に分類することができる。

特に，ソフトウェアの成果物の一部であるシステム設計書，プログラム設計書，フローチャート，ソース・プログラム，などのプログラム関連資料については，比較的容易にその「著作物性」が認められる。しかし，機械語で書かれたオブジェクト・プログラムが著作物性を持つか否かについては，議論の分れるところであり，文化庁の著作権審議会報告書では否定の立場をとり，米国では肯定されているようである。

昨今のマイコンブームに伴なう，マイコン専門誌上を賑わしているBASICソース・プログラムは，この側面を示しているといってよいであろう。

(ii) 財産的性格

ソフトウェアは，無形の知的産物として，無体財産権の対象となり得るが，仮りに特許権または著作権の対象としての適格要件を欠いたとしても，相当な経済的価値をもった無形の財産である。経済的価値があるというためには，ソフトウェアは交換価値を持たなければならないが，ソフトウェアのアンバンドリング以来，独立の交換価値が認められるに至っている。(2)

すなわち，こゝでは再びソフトウェアの「物」としての側面が登場する。

ソフトウェアの法的保護を考える場合，結局「ソフトウェアとは何か」という根源の問題にさかのぼらざるを得ず，中間報告書には様々な定義

注(2) (社)ソフトウェア産業振興協会　ソフトウェア法的保護調査研究特別委員会ソフトウェア法的保護の調査研究に関する中間報告書　昭和56年3月（P18～21）

や見解が採録されている。これをみると定説が現われるまでに，まだかなりの年月を要すると思われる。

この問題についての興味深い主張の一つに，コンピュタ関係の法律専門家，米国ボストン大学フリード氏による次の説がある。それは「ソフトウェアという言葉を形容詞として使い，　software program, software document というように，コンピュータ・プログラムの存在形式によって，用語を規定すべきである。」というものである。

たしかに，ソフトウェアをそれの権利保護の面から考えると，ソフトウェアがたとえ仮りの姿だとしても，ある形をした存在として捉え，印刷物，ソーステープ，ファームウェアなど，それらの個々について論じる必要があろう。

(2)　ソフトウェアの特性

法人税基本通達８－１－７（昭和５５.５.１５ 改訂）に，ソフトウェア開発費用について，次のような説明がなされている。

「他の者からソフトウェアの提供を受け，又は他の者に委託してソフトウェアを開発した場合にその提供を受ける，又は委託するために要した費用は，令１４条第１項第９号ハ《役務の提供を受けるために支出する権利金等》に規定する繰延資産に該当する。」

さらに，法人税法施行令第１４条が規定する繰延資産の範囲については，①創業費，②建設利息，③開業費，④試験研究費，⑤開発費，⑥新株発行費，⑦社債発行費，⑧社債発行差金，⑨その他支出の効界が支出の日以後一年に及ぶもの，となっており，ソフトウェアはこの⑨に該当する，としている。

基本通達には，派遣によるコーディング作業を除外しているが，資産増加を避けるために，この派遣コーディングを促進することが考えられる。この場合，ソフトウェアに対する社会的責任の増大に伴って，ソフトウェアの開発が素人の手作りになり勝ちな自社開発指向から，徐々に成果物に責任を持つ請負契約に変りつゝあった時に，その傾向にブレーキをかける取扱いが打出されたことについては，今後の影響が注目されるところである。

ソフトウェアの捉え方は，このように多様であるが，その原因について考えてみると，次のようにまとめられる。

(a) 過去に類似のものが存在しない。

あるいは少しづつ似ているものが多くある，と言うべきかもしれない。そして，この故に，多様な捉え方が生れている。

(b) 範囲が広い。

ハードウェアの対照語から派生したために，ハードウェア以外のすべて，を捉えることは僅く自然である。このため，互いに異った意味でソフトウェアという言葉を用い，混乱を招くことがよくある。

○目的や用途に従って，表現形態が変遷する。

プログラム設計のためにシステム設計書があり，プログラミングのためにプログラム設計書がある。プログラマがコンピュータに読ませるためにソース・プログラムを書き，コンパイルの結果作られるオブジェクト・プログラムはコンピュータのメモリに読ませるための形態である。実際に稼動中のソフトウェアは，コンピュータの機構が受け付け得るメモリ上のオン・オフ信号の集積である。

問題を複雑にする原因の一つは，これらすべてをソフトウェアと呼ぶことにある。

(d) 未だ激しい技術変革のさ中にある。

かってメインメモリ上で動いたソフトウェアは，今はROMやマイクロプロセッサなど，ICやLSI上で動くようになった。

作り方も大きく変りつゝある。例えば業務プログラムの分野では，プログラマでない一般の人々が使えるプログラミングシステムが使われ始めたが，一般の人がマシンに与えるコマンド群は，今まで扱ってきたソフトウェアとはかなり異質なものとなる。

### 2.1.1 ソフトウェア分類の考え方

分類には2つのアプローチがある。一つは対象に対す見方を整理するためのものである。

6つの側面から眺めた図面から，立体を表現できるように対象を幾つかの性格を異にする側面，すなわち属性で捉え，個々の属性を定義すると同時に，同じ属性に属する区分を定義する。通常区分は時代の変化に伴なって変化するが，属性は不変である。

もう一つは，個々の対象を箱に入れるための分類である。統計などの実務では，これは欠かせない。そして，この分類を決めるに当っても，前者の属性の整理は極めて有効である。

特に，前述のように様々な捉え方がなされ，しかも今なお日進月歩であるソフトウェアに対しては，属性の整理から着手することが妥当であろう。

ソフトウェアの捉え方を大別すると，コンピュータ・プログラムとほゞ同義として捉える狭義のソフトウェアと，ハードウェア以外のサービス，その他すべてを含む広義のソフトウェアがあるが，問題を整理するために，狭義のソフトウェアの属性と，その他のものに対する属性とを区別して扱うことにした。

(1) プログラムとしてのソフトウェア

言語プロセッサという名称には，コンピュータ言語を入力して目的プログラムを出力する，という意味と，コンパイラとしての構造を持つ，という意味が含まれる。前者は取扱かう情報形態を示し，後者は作られ方，つまり内部構造を示す。

また，産業用プログラムという呼び方は，明らかにそのプログラムが製造会社の業務のために作られていることを意味し，適用業種を示す。汎用コンピュータ用，あるいはマイコン用という呼び方は，そのプログラムが稼動する場としてのコンピュータ種別を示す。

このように，よく用いられるプログラムの名称を幾っか挙げてみただけで

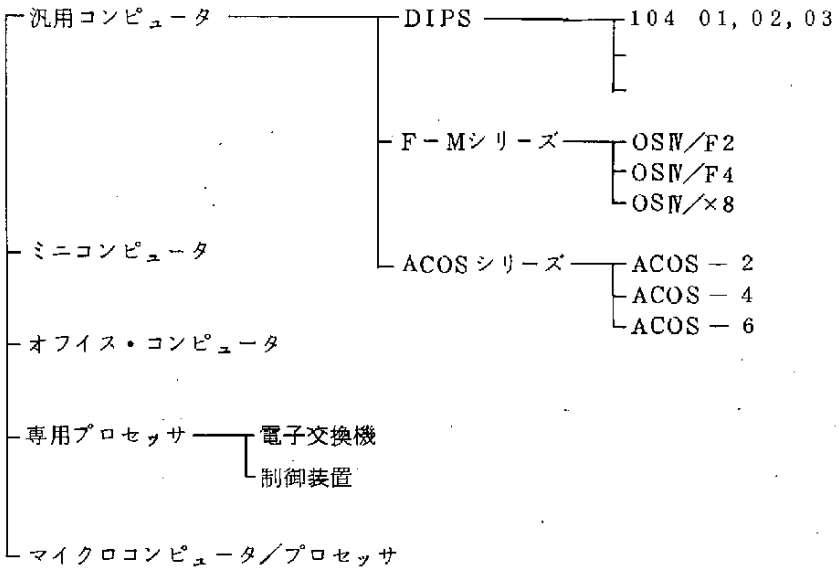
も，これらが全く次元の異なった分類概念の下に分類されていることがわかる。前述の，実務に用いる，箱に入れるための分類では本来の多次元属性を一次元に圧縮せざるを得ないのだが，こゝでは本来の属性にさかのぼって，それを整理してみたい。

(i) 稼動する場からみた属性

ソフトウェアは，発生の当初は個々のコンピュータ個有の附属物であったから，まずそれが稼動する場としての，フォン・ノイマン形ハードウェアによる分類が考えられる。現在，汎用コンピュータは同一アーキテクチャーの下にシリーズあるいはファミリー化されているので，ソフトウェアからみると，同一機種として扱うことができる。

下位属性として，コンピュータの種別より重要なものは，オペレーティング・システム（ＯＳ）である。将来機種を超えたＯＳが普及すれば,OSが上位属性として扱かわれるようになるであろう。

結果として，稼動する場としては，（コンピュータ・アーキテクチャー）＋（ＯＳ）を主な属性と考えることができる。例を示せば次のようになる。



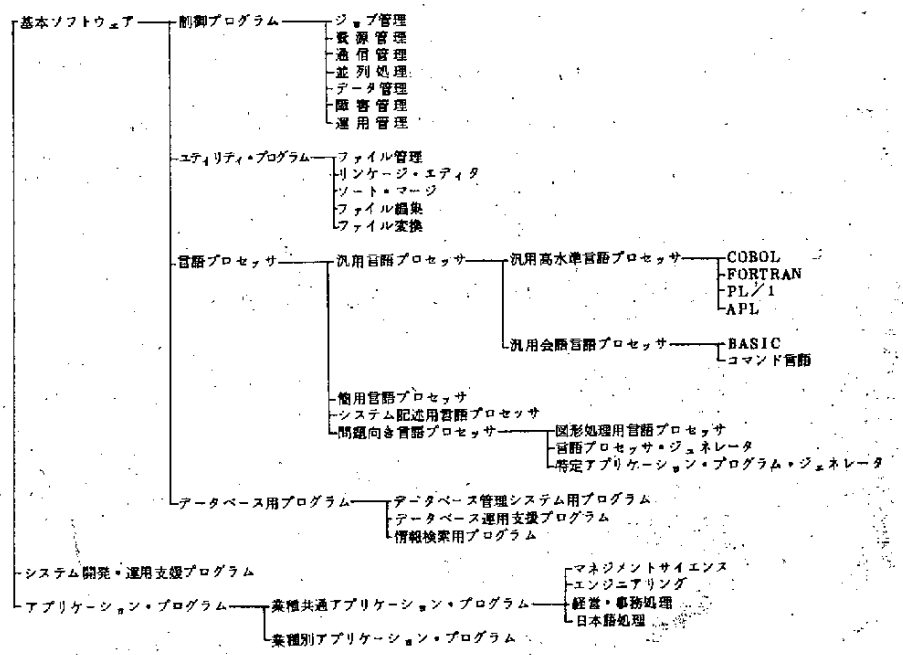
(ii) 機能および構造からみた属性

ソフトウェアの守備範囲からみたのが機能属性である。汎用大型のコンピュータ・システムでは，基本プログラムはあたかも大企業の組織のように，機能が階層的，有機的に構成されており，その外側にアプリケーション・プログラムが位置づけされる。

個々の機能について，通常個有の作成技術や，特有の構造が対応するので，機能属性は概ね構造属性として捉えることもできるが，同一機能であっても言語プロセッサなどでは異なった技術体系に属するものがある。

個有の技術を著積し伝承すべき領域として，この属性は重要な意味を持つ。

この属性は，次のように展開できる。



(iii) システム形態からみた属性

よく用いられるが，例えばネットワーク・システム用プログラムのように，単独で用いた場合は範囲が広過ぎて誤解を招く恐れがある。常に他の属性と組合せて用いるべきであろう。

例えば，次のように展開できる。

- マルチプロセッサ用
- コンピュータ・ネットワーク用
- 分散処理用
- 会話型システム用
- オンラインシステム用
- バッチシステム用

(iv) 直接制御する対象からみた属性

そのプログラムがプロセッサの外に向って何を直接制御しているかを示す。特にハードウェアを制御対象とする場合この属性展開は，よく用いられる。

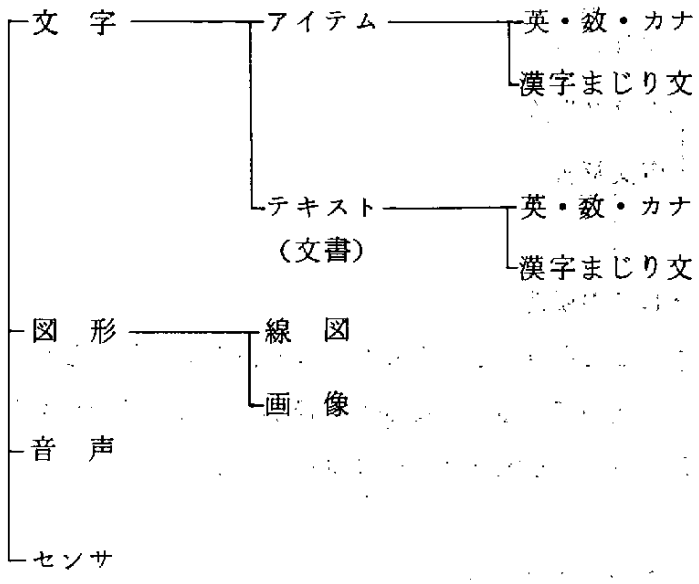
これは，次のように展開できる。

- 他のＣＰＵ
- 入出力機器
- 通信制御装置
- ファイル
- プラント
- ＮＣ装置
- 単体ターンキー・マシン

(V) 扱かう情報形態からみた属性

漢字，図形，音声，画像と取扱う情報形態が拡大するにつれて，こうした属性からもソフトウェアを捉える必要が生じた。

これは，次のように展開できる。




(VI) 対象業務からみた属性

アプリケーション・プログラムに於ては，これが目的であるため，極めて重要な属性である。

これは，一般に用いられる業種——業務——サブ業務（例えば産業用—生産管理用——工程管理用）という属性展開と共通する。

(VII) 特殊な理論や処理技法による属性

適用している理論や処理技法が特に重要である特殊アプリケーションプログラムに於ては，技法が属性となる。

例えば，次のようなものがある。

- ソフトウェア・シミュレータ
- PERT／CPMプログラム
- 数理計画法
- 有限要素法
- 数値計算
- 統計計算
- グラフ理論
- 構文解析
- ⋮

(viii) 商品としての属性

商品にすることを意図して作ったものであるか否か，どういう販売上の性格をもった商品であるかは契約上の扱かいに大きく影響を与える。

現在定着しつゝある区分は，次の通りである。

- バンドル・ソフトウェア
- アンバンドル・ソフトウェア
  - レンタル／リース
  - 売切り，保守付き
  - 売切り，保守なし
- カスタム・ソフトウェア

(ix) その他の属性

以上の他にも幾つか挙げられるが，これらは以下に列記するに止める。

○ 記述言語からみた属性

○ 扱かう情報ファイルからみた属性

例えば

環境情報システム

科学技術文献情報システム

特許情報システム

労働情報システム等

(2) プログラムに関するドキュメント及び媒体

広義のソフトウェアと狭義のそれとの差は，ノウハウ，サービスなどの無形なものと，書類や媒体に記録されて，何等かの形態をもっものとに分けられる。

無形なものは議論の対象から外し，有形のものについて属性を考えると，一つは記録されている内容であり，もう一つは記録されている形態である。これら2つの属性を組合せると，およそ次のようになる。

| 形態(注) / 内容 | 紙 | 印刷物 | 磁気テープ | 磁気ディスク | ROM | LSI回路 |
|---|---|---|---|---|---|---|
| 設計ノウハウ | ドキュメント | 専門書 | | | | |
| システム仕様 | 〃 | | | | | |
| 設計仕様 | 〃 | | | | | |
| 使用法 | | アニュマル | | | | |
| ソース・プログラム | コーディングシート | ソースリスト | ソーステープ | ソースディスク | | |
| オブジュクト・プログラム | | オブジュクトリスト | オブジュクトテープ | オブジュクトディスク | ロードされたROM | ファームウェア |
| データベース | | ○ | ○ | ○ | | |

(注) 形態と，それに入れる内容は，技術進歩によって今後大いに変化し得る。

前述のように，法律家の「ソフトウェアを形容詞として用いるべきだ」という主張は，このマトリックスの各々のエントリについて，個別に法的取扱かいを議論し規定すべきことを意味している。

最近の法的保護の論点は，著作権によるソフトウェアの権利保護が，マトリックスのどの部分までカバーすべきか，ということと，ソフトウェアの特許性の有無，あるとすればそれが成立する条件について，である。

(3) まとめ

以上述べたように，ソフトウェアは多次元の属性空間の中にあり，それら個々の次元が他の分野に若干の類似性を持つ。このことを認識した上で，実務上は実際の頻度や重点に照らして，現実的な捉え方をするべきであろう。

### 2.1.2 「プログラム開発に関する新高度化計画」におけるソフトウェア（プログラム）とは

ソフトウェアとは，2.1.1のソフトウェア分類・体系に記されているように物として定着し得ない又ソフトウェアの属性も種々の立場があり，定まった見方ができるものではない。そこで当委員会としては，コンセプト面，生産実積予測面等から種々検討された結果，通商産業省の「電子計算機利用高度化計画」[(3)]における情報処理の振興を図るため，開発を特に促進する必要があるプログラムの開発目標及び情報処理の振興を図るため，利用を特に促進する必要がある電子計算機の設置目標について定めたものからプログラム開発目標で使用された分類体系に従うことにした。

(1) プログラム開発に関する新高度化計画

この計画は，大きく分けて2つに分かれており，まず1番目について，今後5年間にわたる情報処理量の飛躍的増加，社会・生活領域の情報化，分散処理の進展，プログラムの開発・保守コストの増大等情報処理をめぐる環境の変化を踏まえた新計画の基本理念について述べ，開発すべきプログラムを明記するとともに，ユーザインタフェース，通信プロトコル，プログラム言語等の標準化を推進し，プログラムの流通基盤を整備することに重点を置くことを明言している。

2番目については，具体的に各プログラムにおいて開発すべき事項を述べている。前回の計画と大きく異なる点は，次の2点である。

(i) 前回計画は，開発すべき事項として，単に性能向上，汎用化，実用化の3種類に類別していたが，今回の計画では，さらにその内容を操作性の向上とか信頼性の向上というように，各プログラム毎に開発すべき事項を具体的に明記することによってわかりやすくなったこと。

注(3) 昭和55年11月以来，会長代理野見山勉（情報処理振興事業協会理事長）にハードウェア専門委員グループ（主査宮川公男一橋大学教授）及びソフトウェア専門委員グループ（主査米田信夫東京大学教授）を設けて検討を進め6月30日同審議会に諮り，7月18日告示

(ii) 近年，大量情報の効率的管理の必要性の増加を考慮して，5としてデータベース用プログラムを大項目として追加したこと。

以上の2点が全体として大きく変わった点であるが，細かい変更点については，後に詳細説明を行う際に述べることとする。

全体のプログラム高度化計画は図2－1，図2－2に示すように制御プロセッサ，システム関係，運用支援プログラム，データベース用プログラム及びアプリケーション・プログラムに分けられており，それぞれのプログラム説明は次のようになる。[4]

1. 制御プログラム

コンピュータが効率的に運転されるためには，中央処理装置，入出力装置記憶装置などを，ジョブに合わせて，量的あるいは時間的に最適に割り当てる必要があり，また利用者に対しても容易に利用できる手段が提供される必要がある。制御プログラムとは，このようにコンピュータの全体の動きを統括するプログラムである。

現在のコンピュータシステムは，バッチ処理，オンラインリアルタイム処理，タイムシェアリング処理等を同時に行うことや，複数のコンピュータを結合して利用することが盛んになっており，制御プログラムに要求される機能が多様化してきている。このため，制御プログラムが巨大化し，処理効率や操作性が低下するという弊害が出てきている。

制御プログラムは，全体として，高価なハードウェアを如何に効率良く利用するかということに主眼を置き，ソフトウェア資産の利用を前提とした汎用的な制御プログラムと，近年，マイクロコンピュータ等ハードウェアの価格低下に伴い，特殊用途を指向した専用的な制御プログラムの大きく分けて2つの流れがある。

前者の汎用制御プログラムにおいては，バッチ処理やＴＳＳ処理等を一体化したものが主体となっており，かなり性能向上が図られたが，今後，更に

注(4) 電子計算機利用高度化計画関係資料集　昭和５６年７月通商産業省より抜粋

## 図２－１　電子計算機利用高度化計画

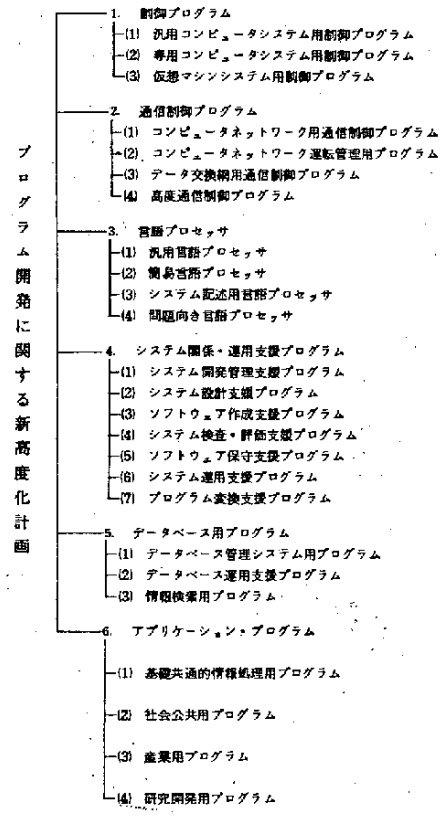


図2－2

プログラムの開発に係わる電子計算機利用高度化計画

1. 制御プログラム
- (1) 汎用コンピュータシステム用制御プログラム
  コンピュータシステムの大規模化，オンライン化，利用者層の拡大等に対処するため，機密保護機能，信頼性，保全性及び操作性の向上，ファームウェア化による処理効率の向上，巨大化したプログラムのモジュール化，分散化等を図る。
- (2) 専用コンピュータシステム用制御プログラム
  周辺機器制御，並列処理，データベース管理，ソフトウェア開発等の特定の用途に用いられる専用コンピュータシステムの利用の高度化を図るため，機能分担及び負荷分担の最適化，マンマシンインタフェースの改善による操作性の向上等を図る。
- (3) 仮想マシンシステム用制御プログラム
  コンピュータシステムを仮想化することによりプログラムの流通促進及び長寿命化を図るため，仮想制御機能の向上，ファームウェア化による処理効率の向上等を図る。

2. 通信制御プログラム
- (1) コンピュータネットワーク用通信制御プログラム
  電子計算機，端末機，回線網等のネットワーク資源の統一的利用を促進するため，ファイル転送，ジョブ転送等における高位プロトコルの整備に即応しつつ，処理効率及び機密保護機能の向上，マンマシンインタフェースの改善による操作性の向上，ネットワークアーキテクチャの異なるコンピュータネットワークを接続するためのプロトコル変換機能の向上等を図る。
- (2) コンピュータネットワーク運転管理用プログラム
  コンピュータネットワークの拡大に伴い，ネットワークを効率良く運転管理するため，ネットワーク資源の最適化管理機能及び遠隔保守機能の向上，ネットワークの自動運転機能及び異常対処機能の向上等を図る。
- (3) データ交換網用通信制御プログラム
  データ交換網の高度利用を推進するため，交換機能の向上，付加機能及び網管理機能の充実当を図る。
- (4) 高度通信制御プログラム
  文字，音声，図形，画像等の多様な情報形態及び構内通信等の多様な通信形態において，通信を効率良く行うため，多様な情報形態を体系的に扱うことのできる通信プロトコルの整備に即応しつつ，情報形態相互の変換機能の向上，高速・広帯域伝送制御機能，通信管理機能の向上等を図る。

3. 言語プロセッサ
- (1) 汎用言語プロセッサ
  プログラム言語の汎用化及び標準化を推進するため，国際標準の動向に準拠した言語プロセッサの開発を促進するとともに，対話機能の実充等による操作性の向上，処理効率の向上，処理内容の高度化等を図る。
- (2) 簡易言語プロセッサ
  一般ユーザによる電子計算機の利用に容易にするため，対話機能及びガイダンス機能の充実等による操作性の向上，適用範囲の拡大等を図る。
- (3) システム記述用言語プロセッサ
  システムソフトウェアの開発を効率良く行うため，システム記述用言語の高水準化，生成プログラムの処理効率の向上等を図る。
- (4) 問題向き言語プロセッサ
  特定の分野におけるプログラムの作成を容易にするため，対話機能及びガイダンス機能の充実等による操作性の向上，処理効率及び図形処理機能の向上，適用分野の拡大等を図る。

4. システム開発・運用支援プログラム
- (1) システム開発管理支援プログラム
  システムの開発工程，ソフトウェアの履歴等を一貫して効率良く管理するため，プログラム，ドキュメント，管理情報等のデータベース化，要員，資金等に関する有効な計画・管理機能の実用化，マンマシンインタフェースの改善による操作性の向上等を図る。
- (2) システム設計支援プログラム
  ハードウェア，ソフトウェア，データベース等のシステム構成を効率良く設計するため，対話機能の充実等による操作性の向上，日本語処理機能及び要求分析・要求定義機能の実用化等を図る。
- (3) ソフトウェア作成支援プログラム
  ソフトウェアを効率良く作成するため，プログラムの自動作成機能の向上，対話機能の充実等による操作性の向上，プログラムモジュールの管理・編集機能の向上，テストデータ作成機能の向上等を図る。
- (4) システム検査・評価支援プログラム
  完成したシステムの検査及び評価を適正かつ効率良く行うため，客観的な検査及び評価の基準の整備に即応しつつ，定量的検査機能の向上，処理能力の測定機能の向上等を図る。
- (5) ソフトウェア保守支援プログラム
  完成したソフトウェアを効率良く保守するため，プログラム解析機能の向上，プログラムの一部変更に伴う影響の解析機能の実用化，ドキュメント編集機能の向上，対話機能の充実等による操作性の向上等を図る。
- (6) システム運用支援プログラム
  システムを効率良く運用するため，自動運転機能の向上，システム運転情報の管理・分析機能の向上，負荷に応じた運転構成の決定機能の向上等を図る。
- (7) プログラム変換支援プログラム
  プログラムの流通促進及び有効利用を図るため，変換率及び変換検証機能の向上等を図る。

5. データベース用プログラム
- (1) データベース管理システム用プログラム
  複雑かつ大量のデータを有機的に収容し，また，巨大化し，分散化し，多様化するデータベースを効率良く管理するため，マンマシンインタフェースの改善による操作性の向上，アクセス手法の改善による処理効率の向上，機密保護機能の向上等を図る。
- (2) データベース運用支援プログラム
  データベース管理システムを効率良く運用するため，データベース作成・再構成機能の向上，データベース間のデータ変換機能の向上等を図る。
- (3) 情報検索用プログラム
  情報検索を容易に行うため，検索効率の向上，マンマシンインタフェースの改善による操作性の向上等を図る。

6. アプリケーションプログラム
- (1) 基礎共通的情報処理用プログラム
  電子計算機の利用の容易化及び高度化を図るため，パターン情報処理機能の向上，日本語，図表等に係る文書処理機能の向上，自然言語解釈，推論等の知識情報処理機能の実用化，機密保護機能の向上等を図る。
- (2) 社会公共用プログラム
  国民生活の利便の向上に資するため，医療用プログラム，教育用プログラム，環境管理用プログラム，防災・防犯用プログラム，生活情報サービス用プログラム等の開発を促進するとともに，マンマシンインタフェースの改善による操作性の向上，機密保護機能の向上等を図る。
- (3) 産業用プログラム
  産業の各部門における業務の高度化に資するため，経営計画・経営管理用プログラム，設計・生産用プログラム，流通・サービス用プログラム等の性能向上を図るとともに，データベース利用機能の向上，マンマシンインタフェースの改善による操作性の向上，業務内容が定型化した分野におけるプログラムの汎用化等を図る。
- (4) 研究開発用プログラム
  開発計画の立案，開発の実施，開発成果の評価等の各段階において，研究開発を効率良く総合的に行い，また，技術先端分野における研究開発を促進するため，実験支援・解析用プログラム，シミュレーション用プログラム等の性能向上を図るとともに，データベース利用機能及び高速計算機利用機能の向上，マンマシンインタフェースの改善による操作性の向上，処理効率の向上等を図る。

利用者の運用，操作性を向上させるために，信頼性，保全性，処理効率を高めるとともに，構成ユニット間の負荷配分や多数の人間が同時に使うため，機密保護機能の充実等が望まれる。

後者の専用的な制御プログラムについては，目的に合ったものが，スタンドアロン型を中心に種々開発されているが，日本語処理による操作性の向上やホストコンピュータと接続した場合の機能分担や負荷分担の最適化機能の向上等を図る必要がある。

また，コンピュータ関連費の中で，ソフトウェアコストが増大している今日，ハードウェアやＯＳが変更される毎にプログラムを作り直すことは困難になりつつある。このため，プログラムのモジュール化や仮想マシンシステムが利用されるようになっており，仮想制御機能や処理効率の向上等が望まれている。

そこで本計画では

(1) 汎用コンピュータシステム用制御プログラム

(2) 専用コンピュータシステム用制御プログラム

(3) 仮想マシンシステム用制御プログラム

に３分類し，それぞれ計画に示されている機能の向上を図ることを目標としている。

2. 通信制御プログラム

通信制御プログラムとは，通信回線と情報処理システムの結合点（フロントエンドプロセッサ等）あるいは，コンピュータネットワークのノード上に位置するプログラムで，コンピュータと端末の間の通信や通信網を有機的かつ体系的に制御・管理するプログラムのことである。

データ通信については，デジタル交換網がすでに実用化され，また通信に際しての発信・受信の規約である通信プロトコルについても国際的な標準化が進み，特に低位プロトコルの標準化はほぼ完成の域に達しているため，効率の良い安定した通信制御プログラムが提供されつつある。

このように通信制御に関する基本的なプログラムは確立されてきているため，開発の重点は，更に高度な分散処理，操作性の向上，パケット交換網の機能充実等に移ってきている。具体的には，ファイル転送，ジョブ転送等の高位プロトコルの整備，異種ネットワークの相互接続機能の向上，文字・音声・画像等の多様な情報形態及び通信形態を体系的に扱うことのできるマルチメディア機能の向上等が望まれている。

そこで本計画では，

(1) コンピュータ間及びコンピュータ，端末間の通信をコントロールするネットワークシステム用通信制御プログラム

(2) コンピュータネットワークの拡大に伴い，ネットワークを効率良く運転管理するためのコンピュータネットワーク運転管理用プログラム

(3) パケット交換，回線交換等のデータ交換網の高度利用を推進するデータ交換用通信制御プログラム

(4) 文字・音声・画像等の多様な情報形態及び光通信，構内通信等の多様な通信形態を体系的に扱うための高度通信制御プログラム

の4つのプログラムに分類し，それぞれのプログラムの開発の目標を定めている。

3. 言語プロセッサ

言語プロセッサとは，FORTRAN，COBOL等の言語をコンピュータが理解できる機械語や中間に介在する言語に翻訳するプログラムのことで，プログラムの開発や実行に際して必要となるものである。

近年の半導体技術の進歩に伴い，ハードウェアの価格は減少しているが，プログラム開発の効率化は進展がみられるものの，ハードウェアの性能向上に比べて進歩が著しく遅い。そこで，COBOL，FORTRAN等の汎用高水準言語プロセッサの処理効率を向上させ，その結果としてできたオブジェクトプログラムの処理スピードを上げるという考え方に基づいて，各種性能向上が行われた。また一方においては，コンピュータの知識をあまり持たない

ユーザが，コンピュータを利用する機会が高まってきているため，簡易で使い易い言語プロセッサの開発が要請されている。

このような状況をふまえ，今回の計画では

(1) COBOL，FORTRANのようにある程度コンピュータに関して知識を持っている人々が使う汎用的な言語プロセッサ

(2) コンピュータに関して，ほとんど知識を持たなくても使うことのできる簡易言語プロセッサ

(3) コンピュータメーカ等においてかなり専門知識を持った人が使う，OS等のシステムを記述するための言語プロセッサ

(4) 実際の利用分野において，各業務に精通したユーザが，特定の問題を処理するために使う問題向きプロセッサ

の4つに分類し，それぞれの達成すべき目標を定めている。

前回の計画と大きく異なる点は，(2)の一般ユーザによるコンピュータの利用を容易にするための簡易言語プロセッサというものを大きく打ち出した点である。

4. システム開発・運用支援プログラム

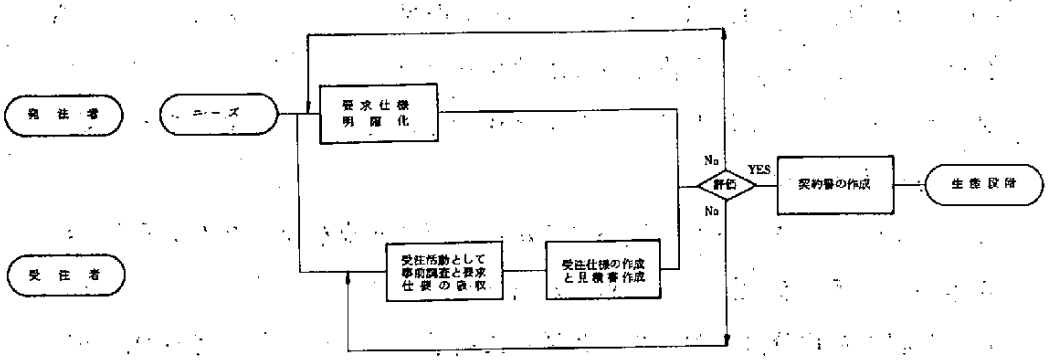
システム開発・運用支援プログラムとは，コンピュータシステムを開発・保守・運用する際に，各種プログラムやドキュメントの開発・保守・管理やシステムの性能評価などを，コンピュータを導入することにより，生産性・信頼性を向上させるためのプログラムである。

過去5年間においては，コンピュータと対話形式でソフトウェアを開発するためのプログラムが充実するなど，ある程度実用レベルのものが開発されたが，まだ性能が十分でなく，限られた範囲の比較的レベルの高い使用者によって利用されているにとどまっている。

近年，ソフトウェアの開発及び保守のためのコストが増大し，今やコンピュータ関連費用のうち，約8割に達していると言われている。（表2－1）

表2－1　ハードウェア，ソフトウェアの開発・保守の費用比率

（TRW社のB.W.ベームによる参考文献）




このため，コンピュータを利用し，ある程度ソフトウェアを自動的に生産することにより，コストを減少させることが必要である。

そこで，ソフトウェア開発を，設計，作成，検査，評価，保守，運用という各段階でとらえ，かつハードウェアも含めたシステム全体として効率良く経済的に開発する重要性が高まってきている。

このような理由から，前回計画では，「ソフトウェア開発管理支援プログラム」となっていたものを，今回はハードウェアとソフトウェアを含めたコンピュータシステム全体としてとらえ，項目を「システム開発・運用支援プログラム」と改め，

(1)　システム開発管理支援プログラム

から

(7)　プログラウ変換プログラム

まで，コンピュータシステムのライフサイクルに沿ってプログラムを整理している。

5. データベース用プログラム

データベース用プログラムとは，大量のデータを有機的に収容しているデータベースを，構築又は利用するためのプログラムで，近年，情報のはんらんと呼ばれる現象のなかで，大量情報を効率的に管理する重要性の増大に伴い，今回新たに大項目としてたてたものであり，このプログラムは今後ともますます重要になるものと思われる。

過去5年間に，コンピュータメーカは各社とも，このようなデータベース用プログラムをＯＳの中に用意し，又，汎用的なパッケージドプログラムの開発も行われ，かなり普及している。

今後は，分散型データベースや大量データベースの構築を目指し，操作性の向上や機密保護機能の向上等を図る必要がある。

6. アプリケーションプログラム

アプリケーションプログラムとは，産業界等において最終ユーザ（エンドユーザ）の問題を，コンピュータを利用して処理する際に使われるプログラムの総称である。

(1)の基礎共通的プログラムとは，文書処理，パターン情報処理，知識情報処理等，プログラムの使われる場所に関係なく，共通的に用いられるプログラムをまとめたものである。

(2)は，医療，教育，環境管理といった非常に公共性の高いプログラムで，前回計画の社会開発用プログラムに対応する。

(3)は，製造部門，流通・サービス部門等の産業分野で用いられるプログラムのことで，ＣＡＤ／ＣＡＭ等の設計・生産用プログラム，配送在庫管理用プログラム，経営計画用プログラム等をまとめたものである。

(4)は，研究分野で用いられるプログラムのことであり，今後，研究開発分野がますます重要になることを予想し，研究作業の初率化，自動化を目指し

たり，原子力・宇宙等の技術先端分野における研究開発を促進するため，スーパーコンピュータの利用等に重点を置いており，前回計画の科学技術計算用プログラムに対応している。

アプリケーションプログラムは，ハードウェアのコストパフォーマンスの上昇，データベースの性能向上，日本語処理の発展等の関連分野の進展と相まって，全般的に実用化及び性能向上が進んだ。しかし，社会公共用プログラムの一部と生産用プログラムの一部に関しては，十分な開発が行われておらず，また，重複開発が目立っており，汎用化が進んでいない。

今後，アプリケーションプログラムの発展を促進するためには，質の良いプログラムを開発すると同時に，業務内容が定型化した分野においては，プログラムのパッケージ化を行い，汎用性を持たせ，プログラムの流通を促進することが重要である。

以下，巻末に詳細Ｖ資料（１２７頁より）参照

## 2.2　ソフトウェアの標準化

## 2.2 ソフトウェアの標準化

ここでは，2.1において限定したソフトウェアに関して，いかなる項目を標準化すべきかを論ずる。標準化すべき項目を，組織的・系統的に導出するために，先ず2.2.1で，何を目的として標準化を考えるのかという根本問題を論じ，2.2.2においてその立場から考察するとき，標準化すべき項目を，どのような場において，どのような面から考えるべきかを論ずる。すなわち，2.2.1及び2.2.2において標準化項目を導出する視座を確立する。

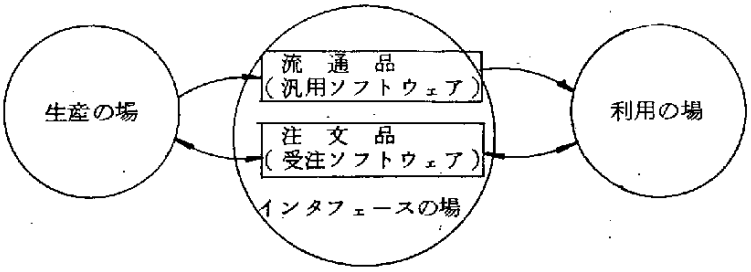
次に，2.2.3において，具体的標準化項目を論じるが，最初に2.2.3.1で，ＩＳＯ／ＴＣ９７及びＪＩＳＣの情報部会においてどのような項目がどのように標準化されているか，また標準化されようとしているか，その現状を述べる。続いて2.2.3.2及び2.2.3.3において，2.2.2で定めた視座から，具体的に必要となる標準化項目を導出し，必要に応じてそれに関して考慮すべき問題を述べる。

### 2.2.1 標準化の目的とその考え方

2.1で述べたように，ここでは製品（プロダクト）としてのソフトウェアを考える。製品には，大きく分けて一般流通品と注文品とがある。これらの製品を媒介として次の三つの場：

- 生産の場
- 利用の場
- 両者のインタフェースの場

が存在する。これらの相互関係を図式的に示すと次のようになる。



さて，製品に対する自然な要求は，利用の場からすれば，良い製品を有効に使用できるということであり，生産の場からすれば，良い製品を効率良くつくるということであり，インタフェースの場からすれば，流通品については，良い製品を選択できるということであり，注文品については，利用者（発注者）・生産者（受注者）間で，製品に関する情報が適確に相互伝達できるということである。すなわち，それぞれの場における上述の目的を整理すると以下のようになる。

(1) 利用の場

(a) 良い製品を入手する。

(b) それを有効（効率良く）使用する。

(2) 生産の場

(a) 良い製品をつくる。

(b) それを効率良くつくる。

(3) インタフェースの場

(a) 一般流通品（汎用ソフトウェア）：良い製品を選択できる。

(b) 注文品（受注ソフトウェア）：発注者・受注者間で情報が適確に相互伝達できる。

標準化の目的は，これらそれぞれの場における目的を実現することであると考える。そこで，ここに示した各事項を更に分析して，どのような項目をどのように標準化したならば，これらの目的が実現されるかを考察する。

### 2.2.2. 標準化項目の体系

前節に示した考え方は，ソフトウェアに限らず工業製品一般に適用できるものであるが，この節では，前節に述べた各目的を一層分析するに当って，製品をソフトウェアに限定した場合について考察する。

前節の(1)a），(2)a），(3)a）においては，いずれも〃良い製品〃ということが問題になっている。また，(3)b）も結局は，注文品に関して〃良い製

品〃を製作し，入手することが目的である。そこで第一に2.2.2.1において，ソフトウェアの場合，〃良い製品〃とは何かを考える。

前節の(1)b），(2)b）は，いずれも〃効率〃を問題にしている。そこで第二に2.2.2.2において，ソフトウェアの場合，〃効率良くつくる〃，〃効率良く使用する〃とは何であるかを論ずる。

最後に，2.2.2.3において前節の(3)の目的が実現されるためには，どのような情報が必要であるかを述べる。

### 2.2.2.1. 良い製品とは何か

(1) 一般流通品（汎用ソフトウェア）の場合

(a) ニーズの充足性：その製品の処理機能が，利用者のニーズをどのくらい満しているかということ。これが判断できるためには，次のことが必要である。

(i) 処理機能の表現法の標準化

(ii) 標準的処理機能の設定

(b) 高性能：処理速度が速いこと及び処理の仕方が良いこと。これが適確に表示されるためには，次のことが標準化されている必要がある。

(i) どのような条件の下で，どのような機能が，どのくらいの速さで処理されるかということの表現法。

(ii) どのような条件の下で，どのような機能が，どのように処理されるかということの表現法。

(c) 高品質：処理機能が保証されていること，オペレーションが容易であること，及び使用法のガイダンスが良いこと。これらのためには，次のことが必要である。

(i) 処理機能が標準化されていること。

(ii) 処理機能のクラス化と，そのクラスごとのオペレーションの標準化。

(iii) ドキュメントが整備されていて，かつ必要な情報が容易・適確に得

られること。

(d) ポータビリティ：ポータビリティは，今日では同種言語異種処理系の場合についていうのが普通である。将来は，システム独立性にまで拡張される可能性はあるが，次のような場合を含めて，先ず次のことが必要となる。

(i) プログラム言語の標準化

(ii) プログラム言語のメーカマニュアルの内容項目の標準化

(e) 導入性：容易かつ適確にシステムへ接続できること及びシステム依存性の少ないこと。このためには，次のことが必要である。

(i) 導入マニュアルが整備されていること

(ii) ＪＣＬの標準化

(f) 管理運用性：操作，管理，保守が容易・適確にできること。このためには，次のことが必要である。

(i) 設計段階に，管理運用性を考慮に入れておくこと。

(ii) 管理運用に関するマニュアルが整備されていること。

(g) 拡張性：量的拡張及び質的拡張が考えられる。このためには，次のことが必要である。

(i) パラメタライズされていること。

(ii) 機能単位が標準化されていること。

(iii) 拡張に関するマニュアルが整備されていること。

(h) 順応性：外的環境条件の変化に対して，どれだけ順応できるか。このためには，次のことが必要である。

(i) 設計段階において，順応性を考慮に入れておくこと。

(ii) 順応性に関するマニュアルが整備されていること。

(i) 価格とのバランス：製品の良さは，製品の能力と価格とのバランスということも問題になる。

(2) 注文品（受注ソフトウェア）の場合

(a) 要求充足性：要求仕様とそれの実現度を評価すること。これのためには，次のことが必要である。

(i) 仕様と実際との差の検査法

(ii) 検査結果の評価法

(b) 価格とのバランス：(1) i ) と同様

2.2.2.2. 効率良くつくる及び使用するとは

(1) 生産の場における効率

生産の場には，設計面及び製造面がある。効率良く生産するとは，品質の保証されたものを，いかに安くつくるかということである。この目的を実現するために，設計面及び製造面でそれぞれでのようなことが考えられるかを以下に述べる。

(a) 設計面：先ず，設計法が組識立てられていることが必要であるが，その場合，設計要求に対する順応性，設計の完全性，設計品質の保証，非熟練者でも一定水準の設計が保証されることなどが実現される手段を考える必要がある。このためには，次のような事項が考えられる。

(i) 設計方法論の整備

(ii) 標準的機能単位の整備

(iii) 設計表現法

(iv) 設計 tool の整備

(v) 設計情報のＩＲシステムの整備

(vi) 設計原価計算法の確立

(b) 製造面：製造管理法が確立していることが必要であるが，その場合，設計者と製作者との情報伝達が適確にできること、非熟練者でも一定水準の製品ができること，作業者交代が可能であること，省人力などが実現される手段を考える必要がある。このためには，次のような事項が考えられる。

(i) 機能単位を用いたアセンブル製造法

(ii) 各製造工程における検査項目

(iii) 製造 tool の整備

(iv) 製造原価計算法の確立

(2) 利用の場における効率

利用の場において，効率が良いとは，購入した製品が，ニーズを充足し，高品質，高性能で，かつ使い易く，しかも能力に比して価格が安いということであろう。これらのうち前3者は，すでに 2.2.2.1 (1) a），b），c）において論じた。そこで，ここでは使い易さにはどのような面があるかを述べる。

(a) ドキュメントの整備：これは更に細かくみると，体系面，内容面，IR面，記述面があり，これらの各面が整備されていることが必要である。

(b) 導入性：これは 2.2.2.1 (1) e）と同様である。

(c) 管理運用性：これは，2.2.2.1 (1) f）と同様である。

(d) 拡張性：これは，2.2.2.1 (1) g）と同様である。

(e) 順応性：これは，2.2.2.1 (1) h）と同様である。

(f) 契約面：使い易さということには，トラブルが起った場合を含めて、アフターサービスの問題も含まれる。

(g) 価格及び経費：購入価格に関しては，2.2.2.1 (1) i）と同様であるが，購入したソフトウェアを走行させる場合の経費も問題になる。

2.2.2.3. インタフェースの場で必要な情報

(1) 一般流通品（汎用ソフトウェア）の場合

利用者が，複数個の商品の中から，どれか一つを選ぶ際に，商品に関してどんな情報があれば選択の判断を下せるかを考える。

(a) 適用分野：その製品が有効に利用できる分野・業務・処理対象が何であるか，また有効でないそれらは何であるかについての情報が必要である。このためには次のことが必要である。

(i) 適確な情報が伝達されるために，用語とその定義が標準化されていること。

(b) 製品機能と入出力：製品の機能表示，必要な入出力情報が必要である。このためには，次のようなことが標準化されている必要がある。

(i) 機能単位・用語

(ii) データの記述

(iii) 属々用いられるファイルの名前及びその内容

(iv) 機能単位をクラス分けし，クラスごとの出力形式

(c) 製品構成：製品としてどのような機能単位が，どのように構成されているかを示す。この場合次のことが必要である。

(i) 機能単位・用語の標準化

(ii) 機能単位の構成表示法

(d) 性能：これは，2.2.2.1(1)b）と同様である。

(e) 品質：これは2.2.2.1(1)c）と同様である。

(f) 製品の物理的構成：製品媒体及びモジュールの物理的構成を示す。これには次のことが必要である。

(i) 媒体及び関連ハードの標準化

(ii) 物理的構成表示法

(g) 環境条件：その製品を使用する際に必要な処理系に対する制限事項を示す。すなわち，入出力データ，ソフトウェア上の条件，ハードウェア上の条件である。このためには，次のことが標準化される必要がある。

(i) 分野別に必要データ及びその名前

(ii) ハードウェアに関するシステム表現法及び機器名

(iii) ＯＳの機能と名前

(h) ポータビリティ：これは，2.2.2.1(1)d）と同様である。

(i) アフターサービス：これについては，次のことが必要である。

(i) サービスの種類の定義と用語の標準化

(j) 価格又は契約条項：価格については，2.2.2.1(1)i）と同様。

(i) 分野別に，標準ソフトウェアをえらび標準価格的なものを公示する。

(ii) 契約条項を標準化する。

(2) 注文品（受注ソフトウェア）の場合

受注ソフトウェアの場合，発注者と受注者間の情報の伝達が本質的である。受注ソフトの場合は，製品が受注されてから納入され，かつ実際に使用される過程を，いくつかに段階を分けて考える。

(a) 発注段階：ここでは，要求仕様の表現法，発注者・受注者間の情報伝達法，契約条項が問題になる。そこで次のようなことが標準化されることが必要である。

(i) 要求仕様言語

(ii) feasibility studyから発注時までの，発注者・受注者間情報伝達法

(iii) 契約条項

(b) 生産段階：ここでは，次のようなことが必要となる。

(i) インタフェース面からみての生産工程の標準化

(ii) 各工程での要求実現度の評価法

(c) 導入段階：ここでは，次のようなことが必要となる。

(i) 導入手順の標準化

(ii) 要求実現度の評価法

(iii) 事前サービスの種類

(d) 運用段階：ここでは，次のようなことが必要となる。

(i) 操作に関するトラブル処置の手順

(ii) 保守に関する手順

(iii) 要求実現度の評価法

(e) 改良・拡張段階：環境変化に対する順応，また量的・質的改良を実施したい場合，次のようなことが必要である。

(i) マニュアル整備

(ii) 改良・拡張の実施法の手順

### 2.2.3　ソフトウェアに関する標準化項目

ここでは，2.2.2で示した視座から，一層具体的に必要となる標準化項目を導き，必要な場合はそれに関して考慮すべき事項を述べる。

このようなことを行うに当って，先ずＩＳＯ及びＪＩＳＣ（日本工業標準調査会）において，ソフトウェアに関ししてどのような項目が規格化されているか，また，されようとしているかを知ることが必要であると考え，2.2.3.1においてこれについて述べる。

次に，2.2.3.2において具体的にソフトウェアに関して標準化すべき項目を論ずる。

#### 2.2.3.1　ＩＳＯ，ＪＩＳＣにおける標準化の現状

(1)　ＩＳＯの現状

ＩＳＯでは，ＴＣ９７が情報処理に関する問題を担当している。特にソフトウェアに関する問題は，ＴＣ９７／ＳＣ１，ＳＣ５，ＳＣ７が扱っている。すなわち，ＳＣ１は用語を，ＳＣ５はプログラム言語を，ＳＣ７は設計と文書化を担当している。

2.2.2で示したソフトウェアに関する問題は，ＳＣ７のscopeに属するものが多い。そこで，ここではＳＣ７の活動状況を中心に紹介し，ＳＣ５，ＳＣ１については簡単にふれることにする。

(a)　ＳＣ７の現状

ＳＣ７は，そのtitleは何回か変更があったが現在は，"Design and Documentation of Computer Based Information Systems"となっており，幹事国はスウェーデンである。ＳＣ７で過去に制定したＩＳＯ規格は，ＩＳＯ１０２８－１９７３（流れ図記号）及びＩＳＯ２６３６－１９７３（流れ図記号の用法）の二つである。それ以後，ＳＣ７はあまり活発な活動をしていなかったが，１９７５年頃より徐々に活動的となり，１９７８年以後はwork itemも増加し，以前にくらべると格段に活動的

になった。

SC7の中に設けられているWGは，次の四つである。

WG 1：Symbols, charts and diagrams

幹事：イギリス（BSI）

WG 2：Items for documentation

幹事：イギリス（BSI）

WG 3：Program design

幹事：カナダ（SCC）

WG 4：Decision tables

幹事：フランス（AFNOR）

また，SC7で現在扱っているテーマは，次のようである。左欄は項目番号を示す。

Item 3：Information processing — Guidelines for the documentation of computer — based application systems

Item 6：Program design

Item 7：Decision tables

Item 8：Symbols and conventions for the flow charts, program network and system resources charts

Item 10：Guideline for program documentation

Item 11：Basic constructs for programs and conventions for their use

Item 12：Structures programming design rules

Item 13：Criteria for the evaluation of programs

Item 14：Criteria for the evaluation of programming methods

Item 15：Program structure diagrams

Item 16：Computer system configuration diagram symbols and conventions

Item 17：Guidelines for data ducumentation

次に，上記各 item についての説明を簡単に述べる。

Item 3：これは，1976年頃より原案が審議されてきたものであるが，受注ソフトとしての Computer-based application system を，設計・製作していく各段階で発注者・受注者間で，どのような文書を作成していくべきかを示したものである。現在，DP6592という文書にまとめられているが，これをDISとして処置することが第8回SC7国際会議（1980）で決められている。これらの内容は，2.2.2で述べた受注ソフトに関するインタフェースの場及び生産の場における多くの標準化項目を含んでいる。

Item 6：Program design という titleであるが，ここで実際に扱われたものは，DP6593（Program flow for processing sequential file in terms of record groups）であり，直列ファイルの処理を標準化したものである。これは1978年の第7回国際会議でDISとしての処置をとることが決められた。

Item 7：Decision table であるが，ここで扱われたものは，DP5806（Specification for single-hit decision tables in data processing）というもので，これも第7回国際会議でDISとして処置することが決められた。

Item 8：これは，流れ用に関するISO規格1028及び2636の改訂版である。第9回（1981）国際会議で，DISとしての処置をとることが決められた。これは，1980年にDISにする投票が行われたが，反対の国が多く，1980年

の第8回国際会議，及び１９８１年の第９回国際会議で改訂されたものである。この改訂には，日本からの寄与がかなり多い。１９８１年１２月にＤＩＳとしての投票の対象になる文書が送られてきている。

Item 10：これは，第6回（１９７６）ベルリン国際会議の際に日本が提案したＮ１５７が原案であって，ＤＰ６５９２の完成後に作業する予定になっていたものである。前述のようにＤＰ６５９２が一応の完成をみたので１９８１年の第９回国際会議から，具体的作業に入ることになった。このため，日本から更にＮ２４１としてＮ１５７の改訂版を第９回国際会議に提出したところ，基本的にＮ２４１に沿って今後国際的に検討していくことが決められた。ＳＣ７国内委員会では，具体的例についてＮ２４１を適用して，一層の改良を検討しつつある。また，ドイツ，イギリスも同様の検討をすることになっている。Ｎ２４１の内容は，part Ｉ と part Ⅱ からなり，part Ｉは主として2.2.2におけるインタフェースの場において必要となる情報の項目から成り，part Ⅱは2.2.2で述べた利用の場において必要となる情報の項目から成っている。

Item 11：これに関するＷＰは数年前よりイギリスから提出されていたが，１９８１年の国際会議で，具体的に議論され始めた。これの内容は，プログラムの基本構造を選定し，それの機能と用語を定義しようとするものである。これは，2.2.2.3 (3)で特に必要性を強調した機能単位の一例と解することができる。

Item 12,13,14：これらは，未だ具体的検討の対象となるようなＷＰが提出されていないので，今後の問題である。

Item 15：これは，structured programmingが盛んであった時期のNassi- ShneidermanやChapinの論文及びBurroughs社

の社内規格を参考にして，１９７８年の第７回国際会議に，日本からN１８３として提出したのが始まりである。その後オランダがＮ２３２として同様のものを提出している。しかしこれらを具体的に審議する段階には未だなっていない。

Item 16：流れ図の改訂が議論され始めた頃，日本は特に入出力関係の記号の改訂の必要性をコメントした。第７回国際会議で，Ｎ１６８として流れ図の改訂版が具体的に提出された時，日本はその原案中にprogram network chart及びconfiguration chartを含めることに反対した。それは，これらは〃流れ〃を示すものではなく，configuration chart はむしろハードウェア的なものであるからである。ドイツの代表は，Ｎ１６８ではソフトウェアの立場からみたconfigurationであると主張した。そこで日本としては，configuration をむしろ別規格として定めるべきであることを強調して来た。１９８１年の第９回国際会議でもこの点を力説し，各国もこれを承認し，流れ図中では，configurationという用語を用いないこととなり，system resources chartということが決められた。日本は１９８１年にハードウェア的なcanfigration chantをnew work itemとすることをＴＣ９７に提案したところ，各国の賛成を得て，このitem１６が承認された。具体的原案としては，Ｎ１７５として第７回（ストックホルム）国際会議に，それの改訂版として第８回（ハーグ）国際会議にＮ２０７を日本が提案した。目下，国内委員会でＮ２０７の一層の改良を検討している。

Item 17：これは，データの文書化であって，１９８１年ドイツが提案し，new work itemとして承認されたものである。2.2.2において，これの必要性は指摘されている。

(b) ＳＣ５の現状

ＩＳＯ／ＴＣ９７／ＳＣ５は，プログラム言語の標準化を担当している。ソフトウェアの標準化において，高水準プログラム言語のもつ役割は，明らかである。

以下に，簡単にＳＣ５で標準化が行われたもの及び行われようとしているものを示す。

ＦＯＲＴＲＡＮ：１９７２年にＲ－１５３９となっていたが，ＦＯＲＴＲＡＮ７７への改訂が行われ，現在はＩＳＯ１５３９－１９８０となっている。更に改期改正をめざしてＦＯＲＴＲＡＮ８×がＡＮＳＩで作成されつつある。

ALGOL：ALGOL 60の言語と入出力は，FORTRANとともに１９７２年にR（RECOMMENDATIONの略）となっていたが，１９７８年に廃止となった。

ＣＯＢＯＬ：１９７２年に，ＡＮＳ　ＣＯＢＯＬをＲ－１９８９とした。その後改訂が行われ，現在はＩＳＯ１９８９－１９７８となっている。ＩＳＯ及びＣＯＤＡＳＹＬでは，更に次の改訂作業が行われている。

ＰＬ／Ｉ：ＥＣＭＡとＡＮＳＩが協力して原案をつくり，現在ＩＳＯ６１６０－１９７９となっている。

このほか次の言語が審議されつつある。miuimal ＢＡＳＩＣ，ＢＡＳＩＣ，ＰＡＳＣＡＬ，ＡＤＡ，Database，ＡＰＬ，Graphic Kernel System（ＧＫＳ），プロセス制御用語，テキストプロセッシング言語，modified ALGOL60。

(c) ＳＣ１の現状

ＩＳＯ／ＴＣ９７／ＳＣ１は，情報処理に関する用語の標準化を担当している。用語はＩＳＯ規格としては，２３８２という番号がつけられ，分野別にSectionが分けられ，現在SectionＩからXXまで設けられている。用語については，各Section別に審議されるので，規格の段階（WD，DP, DIS, IS)がSectionごとに異なっている。その現状を表2-2に示す。

## 表2－2　ISO用語集の現状

（1980年6月現在）

| Sec. | Title | WP | WD | DP | DIS | IS |
|---|---|---|---|---|---|---|
| 01 | Fundamental terms | | | (rev.) X | | X |
| 02 | Arithmetic and logic operations | | | | | X |
| 03 | Equipment technology (selected terms) | | | | | X |
| 04 | Organization of data | | | | | X |
| 05 | Representation of data | | | | | X |
| 06 | Preparation and handling of data | | | | | X |
| 07 | Digital computer programming | | | | | X |
| 08 | Control, integrity and security | | X | | | |
| 09 | Data communication | | | | X | |
| 10 | Operating techniques and facilities | | | | | X |
| 11 | Control, input-output and arithmetic equipment | | | | | X |
| 12 | Storage techniques and data media | | | | | X |
| 13 | Computer graphics and computer micrographics | | | X | | |
| 14 | Reliability, maintenance and availability | | | | | X |
| 15 | Programming languages | | X | | | |
| 16 | Information theory | | | | | X |
| 17 | Data bases | | X | | | |
| 18 | Remote access and distributed data processing | | X | | | |
| 19 | Analog and hybrid computing | | | | | X |
| 20 | System development | | X | | | |

(2) ＪＩＳＣの現状

我が国では，日本工業標準調査会（Japanese Industrial Standards Committee，ＪＩＳＣと略称）が，標準化の仕事を行っている。ＩＳＯには，１９５２年に加入している。

ＪＩＳＣの下に，情報部会があり，この部会が情報処理関係の標準化を担当している。情報処理関係のＪＩＳは，１９６０年代より今日に到るまでに，約７０種制定され，改正が行われたものも多数ある。その現状を表2－3に示す。

表２－３　情報処理関係ＪＩＳ一覧

| ＪＩＳ番号 | 名　　称 | 制　定 | 改　正 | 確　認 |
|---|---|---|---|---|
| B6325－77 | 数値制御プロセッサの出力－CLDATA | 77.12. 1 | 80. 9. | |
| B6326－80 | 数値制御プロセッサの出力－２０００タイプレコード | 80. 9 | | |
| C0803－71 | 印刷電信機のけん盤配列および符号 | 61.11. 1 | － | 78. 3. 1 |
| C6201－76 | 電子計算機プログラム用言語FORTRAN（水準7000） | 67. 5. 1 | 76.11. 1 | |
| C6202－76 | 電子計算機プログラム用言語FORTRAN（水準5000） | 〃 | 〃 | 〃 |
| C6203－76 | 電子計算機プログラム用言語FORTRAN（水準3000） | 〃 | 〃 | 〃 |
| C6205－72 | 電子計算機プログラム用言語COBOL | 72. 8. 1 | 80. 9 | |
| C6210－72 | 電子計算機プログラム用言語ALGOL（水準7000） | 67. 5. 1 | 72. 3. 1 | 78. 3. 1 |
| C6211－72 | 電子計算機プログラム用言語ALGOL（水準6000） | 〃 | 〃 | 〃 |
| C6212－72 | 電子計算機プログラム用言語ALGOL（水準5000） | 〃 | 〃 | 〃 |
| C6213－72 | 電子計算機プログラム用言語ALGOL（水準4000） | 〃 | 〃 | 〃 |
| C6214－72 | 電子計算機プログラム用言語ALGOL（水準3000） | 〃 | 〃 | 〃 |
| C6215－72 | 電子計算機プログラム用言語ALGOLの入出力（水準70） | 〃 | 〃 | 〃 |
| C6216－72 | 電子計算機プログラム用言語ALGOLの入出力（水準60） | 67. 5. 1 | 72. 3. 1 | 78. 3. 1 |
| C6217－72 | 電子計算機プログラム用言語ALGOLの入出力（水準50） | 〃 | 〃 | 〃 |
| C6218－72 | 電子計算機プログラム用言語ALGOLの入出力（水準40） | 〃 | 〃 | 〃 |
| C6219－72 | 電子計算機プログラム用言語ALGOLの入出力（水準30） | 〃 | 〃 | 〃 |
| C6220－76 | 情報交換用符号 | 69. 6. 1 | 76. 1. 1 | |
| C6221－73 | 情報交換用および数値制御機械用符号のテープ上での表現 | 〃 | 73. 3. 1 | 76. 3. 1 |
| C6222－69 | 情報交換用符号の磁気テープ上での表現 | 〃 | － | 76. 1. 1 |
| C6223－72 | 情報交換用符号の紙テープ上での表現 | 72. 3. 1 | － | 78. 3. 1 |
| C6224－77 | 情報交換用符号の磁気セットテープ上での表現 | 77. 3. 1 | － | － |
| C6225－79 | 情報交換用漢字符号系のため制御文字符号 | 79. 6. 1 | | |
| C6226－78 | 情報交換用漢字符号系 | 78. 1. 1 | － | － |
| C6227－76 | 情報交換用機能キャラクタ図形表現 | 76. 1. 1 | － | |
| C6228－75 | 情報交換用符号の拡張法 | 75. 3. 1 | － | 78. 3. 1 |
| C7229－78 | 光学式文字認識のための情報交換用符号 | 74. 6. 1 | 78. 3. 1 | |

| JIS番号 | 名　　称 | 制定 | 改正 | 確認 |
|---|---|---|---|---|
| C6230−77 | 情報処理用語 | 70.10. 1 | 77.12. 1 | 改正中 |
| C6223−80 | 情報処理系けん盤配列 | 72. 2. 1 | 80. 2. 1 | |
| C6240−70 | 情報交換用磁気テープ | 70. 4. 1 | − | 76. 4. 1 |
| C6241−72 | NRZI方式による情報交換用磁気テープの情報記録様式 | 72. 8. 1 | − | 78. 8. 1 |
| C6242−77 | 情報交換用磁気テープリール | 71. 1. 1 | 77. 3. 1 | 80. 4. 1 |
| C6243−76 | 情報交換用紙テープ | 70. 4. 1 | 76.11. 1 | |
| C6244−75 | 情報交換用紙カード | 72. 2. 1 | 75. 8. 1 | 78. 8. 1 |
| C6245−78 | 情報交換用磁気テープのラベルとファイル構成 | 70. 4. 1 | 78.11. 1 | − |
| C6246−71 | 情報交換用紙テープ孔の位置と寸法 | 71. 3. 1 | − | 80. 4. 1 |
| C6247−73 | 情報交換用紙カードの孔の位置と寸法 | 73. 3. 1 | − | 76. 3. 1 |
| C6248−74 | 6枚形磁気ディスクパックの機械的互換性 | 74. 3. 1 | | 80. 4. 1 |
| C6249−75 | 6枚形磁気ディスクパックの磁気的特性 | 75. 8. 1 | | 78. 8. 1 |
| C6250−76 | 光学式文字認識のための字形（英数字） | 70. 6. 1 | 76.11. 1 | |
| C6251−80 | 磁気インキ文字読取用字体および印字仕様（E13B） | 71. 3. 1 | 80. | |
| C6252−80 | 光学文字認識のための字形（片仮名） | 73. 8. 1 | 80. 2. 1 | |
| C6253−75 | 光学式文字認識のための印字仕様 | 75. 2. 1 | | 78. 2. 1 |
| C6254−79 | 光学式文字認識のための手書き文字（片仮名） | 79.11. 1 | | |
| C6255−79 | 光学式文字の認識のための手書き文字（数字） | 〃 | | |
| C6260−73 | 都道府県コード | 70. 4. 1 | 73. 4. 1 | 76. 4. 1 |
| C6261−80 | 市区町村コード | 〃 | 80. 4. 1 | |
| C6262−77 | 日付の表示（コード） | 〃 | 77.12. 1 | |
| C6263−77 | 時刻の表示（コード） | 〃 | 〃 | − |
| C6264−71 | 性別のコード | 71. 1. 1 | − | 80. 4. 1 |
| C6265−80 | 産業分類コード | 71. 4. 1 | 80. | 76. 3. 1 |
| C6266−80 | 職業分類コード | 72. 3. 1 | 〃 | 78. 3. 1 |
| C6267−77 | 商品分類コード | 74. 3. 1 | 77.12. 1 | |
| C6269−76 | 勘定科目コード | 76. 1. 1 | | |
| C6270−75 | 情報処理流れ図記号 | 70. 4. 1 | 75. 2. 1 | 78. 2. 1 |
| C6271−73 | 2値論理素子記号 | 73. 3. 1 | − | 76. 3. 1 |
| C6272−76 | 電子計算機組織構成機器の性能表示 | 72.12. 1 | 76. 1. 1 | |
| C6273−78 | 情報交換用単位記号 | 75. 3. 1 | 78. 8. 1 | 78. 3. 1 |
| C6280−75 | 情報交換用磁気カセットテープ | 75. 3. 1 | − | 78. 8. 1 |
| C6281−75 | 情報交換用磁気カセットテープの情報記録様式 | 〃 | | 〃 |
| C6282−75 | 位相変調方式による情報交換用磁気テープの情報記録様式 | 75. 2. 1 | − | 78. 2. 1 |
| C6283−75 | 情報処理用連続伝票 | 〃 | − | 〃 |
| C6284−75 | 11枚形磁気ディスクパックの機械的互換性 | 〃 | − | 〃 |
| C6285−80 | 11枚形磁気ディスクパックの磁気的特性 | 76.11. 1 | 80. 2. 1 | |
| C6286−80 | 1枚形磁気ディスクカートリッジ（上面着脱形）の機械的互換性 | 〃 | 〃 | |
| C6290−78 | フレキシブル・ディスクカートリッジ | 78.11. 1 | − | 80. 4. 1 |

| JIS番号 | 名　　称 | 制定 | 改 正 | 確 認 |
|---|---|---|---|---|
| C6301−74 | 学歴区分コード | 74. 3. 1 | − | 77. 3. 1 |
| C6302−78 | 大学・高等専門学校コード | 〃 | 78.11. 1 | 77. 3. 1 |
| C6303−74 | 続柄コード | 〃 | − | 80. 4. 1 |
| C6304−76 | 地域メッシュコード | 76. 1. 1 | | |
| C6360−71 | 伝送回線上のキャラクタ構成と水平パリティの用法 | 71. 4. 1 | | 77. 1. 1 |
| C6361−71 | モデムと通信制御装置およびデータ端末装置とのインタフェース | 71. 4. 1 | − | 77. 1. 1 |
| C6362−75 | 基本形データ伝送制御手順 | 75. 8. 1 | | 78. 8. 1 |
| C6263−78 | ハイレベルデータリンク制御手順のフレーム構成 | 78.11. 1 | − | − |
| C6264−78 | ハイレベルデータリンク制御手順の要素 | 〃 | − | − |
| C6265−78 | ハイレベルデータリンク制御手順クラス | 〃 | − | − |

今までのＪＩＳは，多くの場合，ある項目がＩＳＯでＤＩＳとして承認されると，工業技術院より，その項目についてＪＩＳ原案作成の委託が，情報処理学会規格委員会，(財)日本情報処理開発協会又は（社）電子工業振興協会に出され，原案作成委員会が組織されて原案の具体的作成・審議が行われ，情報部会の承認をへて，制定されて来た。

このような制定手順は，ＧＡＴＴ承認という国際情勢の下では，日本固有の項目を除いて，常にふまなければならなくなるであろう。したがって，ＩＳＯで採用されている項目については，それがＪＩＳになることを前提として，積極的に日本の意見を反映させ，規格作成に参加していく必要がある。

表２から明かなように，2.2.2で述べたソフトウェアに関する項目でＪＩＳとして制定されているものは，現在（１９８１年）ＦＯＲＴＲＡＮ，ＡＬＧＯＬ，及びその入出力，ＣＯＢＯＬ，情報処理用語，流れ図記号だけである。このうち，ＦＯＲＴＲＡＮはその内容をＦＯＲＴＲＡＮ7.7に改めた改正原案が１９８１年に完成し，１９８２年に公表されることになっている。情報処理用語は，１９８１年に大改正が行われ，改正原案作成時（１９７９年末）にＤＩＳ以上になっていたＩＳＯ２３８２の各Section（ただし，Section16を除く）をすべて含むようになった。流れ図記号については，ＩＳＯ／ＴＣ９７／ＳＣ７のItem 8で述べたように，１９８１年末現在，ＤＩＳとして承認される可能性の高い文書が送付されて来ているので，ＪＩＳとしても近い時期に改正原案の作成を開始する必要があるであろう。

(3) 利用者・生産者間インタフェースにおいて必要な標準項目

A 汎用ソフトウェア

歴史的には，汎用ソフトウェアはコンピュータの附属物としてハードウェアとバンドルされたオペレーティング・システムから出発した。独立した商品としての汎用ソフトウェアが本格的に市場に流通し始めたのは，昭和45年のIBMによるソフトウェア・アンバンドリング政策実施以降である。

その後，コンピュータ技術の発展に伴なって，汎用ソフトウェアも多様な発展を遂げたが，それをインタフェース標準化の面から整理すると，次のようになる。

| | 区分 | | 例 | 主な契約形態 |
|---|---|---|---|---|
| A | バンドル・ソフトウェア | | IBM SCP | ハードウェアと一括 |
| B | アンバンドル・ソフトウェア | バージョン，リビジョン，アップを伴なうもの | COBOL MARKⅣ | レンタル／リース 売切り＋保守 |
| C | (プログラム・プロダクト 汎用パッケージ) | 〃 〃 〃 を伴なわないもの | 有限要素法 | 売切り |

上表で，Aはハードウェアの一部としての位置づけであり，Cはノウハウ販売に近く，結局，個々のソフトウェアに関して，利用者と生産者が半永久的に関りを持つ，Bのバージョン・リビジョン・アップを伴なう，アンバンドル・ソフトウェアを中心に，その標準化項目を検討することとしたい。

(i) 商品としてのソフトウェアの特質

ソフトウェアを取巻く環境の面から眺めると，閉ぢた環境と，開いた環境に大別できる。前者は，単一的，ターンキィ的なシステムで，専用的に使われる。この場合のソフトウェアは，たとえ独立商品であってもハードウェア機能をソフトウェアに置き換えたものと考えることができるから，ソフトウェア商品としての特殊性に乏しい。他の商品に比べて際立った特質を持つのは後者である。

汎用コンピュータの世界をリードしてきたIBMは，成長の初期に於

て「マシンでなくサービスを売る」商売をレンタルという制度を通じて行ってきた。コンピュータや入出力機器の発展に伴なって，サービス対象としてのオペレーティング・システム（OS）は，メーカの費用で機能強化が続けられた。これがOSのバージョン／リビジョン・アップである。

独立商品となっても，市場は当然継続的な機能強化を期待し，ソフトウェア・ベンダはソフトウェア製品の保守というかたちでその期待に応え，その結果としてソフトウェア市場が大きく成長した。

保守と呼ばれるものは，ハードウェアのそれと著るしく異なった意味を持つ。ソフトウェアの保守に於ては，故障や瑕疵の補修は僅く一部に過ぎない。その大部分は，使用環境の変化に適応して生命を保つための機能強化である。

通常，汎用ソフトウェアは，数多くの構成要素と共に，システムを構成する。そして，この構成要素の各々は，次のような要因によって，かなり頻繁に変化する，そして市場はそれがこの変化に追随することを望む。

| システム構成要素 | 変　化　要　因 |
|---|---|
| 処　理　装　置 | メモリ増加<br>そのもののグレード・アップ |
| 周　辺　装　置 | 新装置の付加<br>台数の増減 |
| オペレーテイングシステム | 機能拡張（バージョン・アップ）<br>グレード・アップ |
| コ ン パ イ ラ | 機能拡張（バージョン・アップ）<br>グレード・アップ |
| アプリケーション・プログラム | 業務の変更に伴なう改造 |
| 汎用ソフトウェア | 上のすべて，および汎用ソフトウェア自体の瑕疵修正および改良 |

これに対応するために，汎用ソフトウェア・メーカは，汎用ソフトウェアを発売する際に，これらの要因による定期的バージョンアップを投資計画に折込み，さらに，保守契約の代価，あるいはオプション化による別建て価格の設定などの政策を販売計画に折り込む。

この保守費の額は，米国の例では初期開発費の２～３倍と言われており，これが発売後長期にわたって支出される点が，汎用ソフトウェアの大きな特質の一つである。

この保守という名の改良を継続するという特色は，コンピュータの機種交替の際には特別の意味を持つ。すなわち，ハードウェアの世代交替を超えて，汎用ソフトウェアがその寿命を保ち続けることが可能となる。事実，アメリカの成功例では，ＩＢＭ３６０から３７０，３０３０，４３００，および３０８１と４世代にわたって寿命を超えた，半永久的な寿命を保ち得ることも，汎用ソフトウェアのもう一つの特質である。

使用者と生産者の利害に直接関連を持つ，汎用ソフトウェアのこれらの特質を踏まえた上で，以下に両者のインタフェースにおいて必要な情報を，区分毎に述べる。

(ii) 購入希望者が選択するための項目

購入希望者は，ある対象業務について，何等かの目的意識を持って，自らの制約条件の下で適用可能な汎用ソフトウェアを探し，幾っかの候補を選ぶであろう。この際に，少なくとも，適用分野，機能，構成，使用環境，価格とその条件，などについての概略の情報が必要であろう。

これらの情報は，公平を期すために少なくとも項目の標準化が必要である。さらに，汎用ソフトウェアが国際的に流通しようとしている時，この標準化は国際レベルで行われることが望ましい。

昨年ＩＳＯ(注1)のＴＣ　９７(注2)，ＳＣ　７(注3)に於て，今後Program Documentation の標準化を，日本から提出された原案に沿って審議することが採決されたが，この原案のPart Ｉ には，講入希望者が選択するための項目として，次の項目が挙げられている。

1. Identification

1.1 Title and subtitle

1.2 Version and date

1.3 History of current version

1.4 Anthor

1.4 Others

2. Purpose and Field of Application

2.1 Purpose

2.2 Field of application

2.3 Merits

2.4 Others

3. Program Specifications

3.1 Problem

3.2 Problem solution

3.3 Functional specification

4. External Specifications of Data and Files

4.1 Input and output data

4.2 Files

5. Operational Specifications

5.1 Language

5.2 Environment

5.3 Opevating modes

5.4 Operation

5.5 Others

6. Contractual Items

6.1 Modnles supplied and test

6.2 Installation and test

6.3 Training

6.4 Maintenace

6.5 Enhancements

6.6 Compensation

(注4) International Organization for Standardigation

(注5) Technical Committee 97 :

(注6) Subcommittee 7 :

(注7) Japanese proposal for documentation N241

6.7 Security

6.8 Documentation

6.9 Others

また，表現の正確さと公平を期すために，言葉の使用に対して宣伝に走らぬようなある種の規制が必要であるが，少なくとも技術用語に関しては，ＪＩＳ　Ｃ－６２３０　情報処理用語に従って使用されねばならない。ＪＩＳに規定されない技術用語は，説明文脈の中で自己定義されるべきである。

(iii) 購入希望者が購入を決断するための項目

幾つかの候補の中からさらに条件を絞り，購入を決断するためには，前項で挙げた項目の詳細の他に，次のようなことをもっと詳しく知ろうとするのであろう。

○何ができるかは判ったが，果して期待する性能の下でそれが得られるであろうか，……性能

○設計やプログラム上の欠陥に悩まされることはないか……品質

○将来の環境の変化に対応できる設計思想，あるいは構造になっているだろうか……拡張の可能性

○将来コンピュータやＯＳを変えた場合にも追随し易くなっているだろうか……可搬性

○将来の環境の変化（ハードウェア，ＯＳベンダの意志によるもの，及び使用者の意志によるものを含む）に対して，どこまで追随してくれるのか，また，その費用は幾らか……保守サービスの範囲と限界

○どういう契約形態が，費用の面で利用形態に適しているか，……契約条件と価格

これらの各項目に対して，次にそのポイントと問題点を挙げる。

○性能

ソート・プログラムのように，入力と出力の関係が比較的明確なものは，標準的な性能測定法を決めることは可能であろうが，条件の数が多いため，一般には極めて困難である。

しかし，よく使われる条件を仮りに定めてでも，概略の性能は表示すべきである。性能曲線がたとへ巾を持っものになったとしてもグラフで示されることが望ましい。

○品質

数え方と把握の困難さから，この表示も極めて困難である。使用者は，これの代用指標として，販売件数，他の使用者の評判，会社の信用度，などを用いているのが現状である。

現在，アメリカの一部で，テスト・カバレージ比率で品質を評価し，表示することが行われているが，これが普及し，表示されるようになれば，有効な品質指標となるであろう。

○拡張の可能性

前述のように，汎用ソフト品としてのソフトウェアの特質は環境の変化の激しさにあるため，この項目は特に重要な意味を持つ。

一般に，システム・ソフトウェアの変化要因の多くは，ハードウェアやＯＳであるから，その対応策をそれ自体の中に組み込むか，保守契約の中でバージョンアップとして行われることが多い。したがって，拡張の可能性は，インタフェース仕様のフレキシビリテイと，保守サービスの内容によって決まる。

アプリケーション・ソフトウェアは，使用者側の業務に変化要因が存在することが多いため，使用者の手で拡張せざるを得ない場合がある。この場合は，そのソフトウェアを改造するために必要な次のような情報やサービスの有無や良否は特に重要である。

記述ランゲージ

ソース・プログラム・リスト

プログラム・ドキュメンテーション

改造のための説明書

コンサルテーション・サービス

これらのうち，プログラム・ドキュメンテーションについては，受注ソフトウェアと同様，規格に従って記述されたものであることが望ましい。

○可搬性

現時点で，コンピュータ・システムやＯＳをまたがって汎用ソフトウェアを使用する，ということは，あらかじめソフトウェア・ベンダが用意した幾つかのサポート条件の異なるバージョンのうち，適したものを選ぶ行為に等しい。しかも，用意される範囲は，概ね同じあるいは類似のコンピュータ・ファミリィに限定される。

しかし，現在すでに，ＵＮＩＸ，ＣＰ／Ｍにみられる異機種共通に使えるＯＳの普及，異機種間ポータビリティを標榜する記述ランゲージＡＤＡの出現など，コンピュータの機種バウンダリィを越えて，ソフトウェア資産を共有しようとする動きが活潑になりつゝある。

これらのものが，事実上の規格，という段階を経て，正規の規格になればその利益は使用者，生産者の双方にとって，はかり知れない。

○保守サービスの範囲と限界

保守サービスは，生きたシステム環境の中で，汎用ソフトウェアを生きたものとしておくためには不可欠のサービスである。

しかも，生産者にとって，不確実な，しかも他動的なサポート環境の変化に追随することを，あらゆる場合に対して確約することは，コストの面から不可能である。

したがって，通常，保守サービスの内容と，その各々の項目に対

して，保守費（あるいは本体価格の一部）に含まれるもの，有償となるものが明示される。

この表示は，とかく誤解を生じ易いので，可能な限り具体的記述がなされるような規格の設定が望まれる。

○契約条件と価格

汎用ソフトウェアを取引する上で特に注意を要することは，価格に含まれる取引対象が変化に富み，しかもプログラム以外の対象がそれと同じ位重要なことである。

例えば，単なるオブジェクト・プログラム，それにソースプログラムやドキュメントが加わる場合，それらにさらに保守サービスがついた場合，の3つのケースでは，その価格には数倍から十数倍の開きがある。なぜならば，ソースプログラム，ドキュメントは設計ノウハウと同じ価値を持ち，保守サービスはプログラムの効用の持続期間に直接に関わるからである。

(iv) 使用者が使いこなすための項目

導入を決定した後，それをうまく使いこなすために，今まで述べた項目の他に，さらに次のようなものが必要である。

○操作法

通常，オペレータが使用するもので，マニュアル化されている。現在発行されているマニュアルの内容は，生産者によって，精粗，文体，図示法等，千差万別である。これについても，容易に，しかも誤りなく操作を行うことを目的として，少なくとも記載項目，図示法，用語の統一が必要であろう。

アプリケーション・プログラムのうち，データベースを用いてレポートを作成するような比較的単純なものは，非手続型のソフトウェアシステムによって素人が行う傾向が顕在化しており，この傾向が強まれば，オペレーション・マニュアルのウエイトは軽くなってい

くと思われる。

○運用法

汎用ソフトウェアを正しく使用するために，ファイルの世代管理，データベースの更新など，あらかじめの環境を整えておく必要がある。このような，コンピュータ・システムを運営管理するマネージャ向きの仕事を円滑に行うために，運用マニュアルが必要であろう。

そして，これらは，コンピュータ自体，あるいは周辺器機に固有の媒体等に対する運用管理法と併せて利用されねばならないが，この面は，例えばＭＳＳの運用法に適切なものがないように，現在のところ著るしく不備の状態にあると思われる。この面の標準化が望まれる。

○改造のための詳細

実際に使用者が改造しようとする場合，その機能に関わる部分の詳細情報が必須である。そして，これは使用者，生産者の両者にとって，必要最少限の詳細情報であることが望ましく，このためには，起り得る改造ケースを予想し，これらの各々の場合についての改造手引書が作られていることが望ましい。

B　受注ソフトウェア

ユーザの依頼に依ってソフトウェアを生産する場合，発注者側との間にはソフトウェア・プロダクトを中心に開発工程全般にわたって種々の情報が往来し，分析・整理されて最終的にはプログラムが納品されることになる。

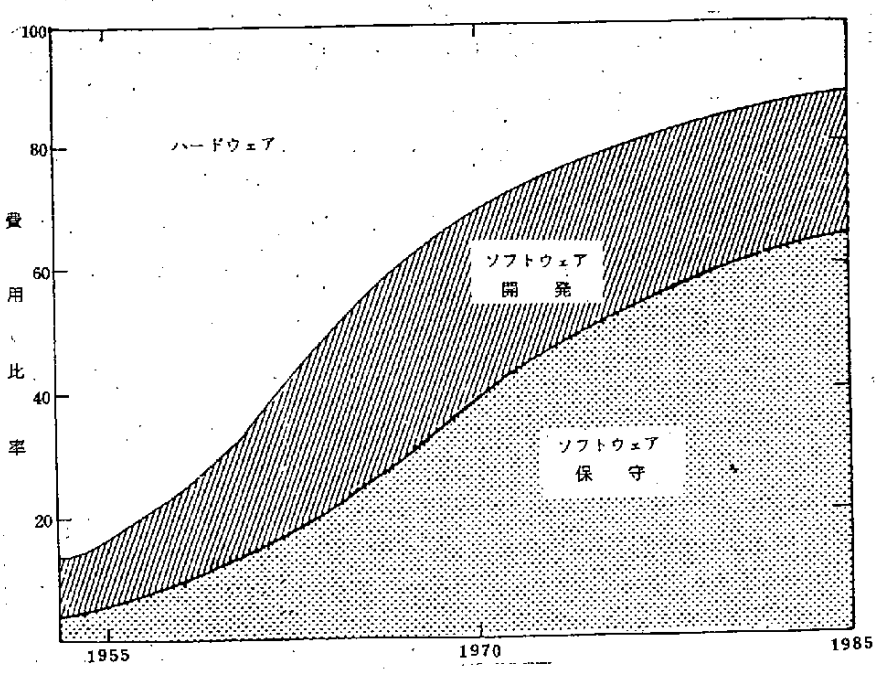
この場合，情報伝達がスムーズでしかも正確に行われない限り，良いソフトウェア・プロダクトとなり得ない。そこでこの受注段階に於いてソフトウェア生産上如何なる情報を必要とし，しかも如何なる標準化を必要とするかを検討する。

(i)　発注段階

この段階は，色々な事柄が不透明で必ずしも明確になっていない場合が多い。この段階で重要なことは，発注者側の意図している内容やニーズを明確にすることシステム範囲を確定して見積を如何に正確に行うかである。この場合受注者側でこの作業に携わる要員は，企業毎の組織体制や業務処理手順などに依って異なるが，営業部門が独立していてプロジェクト規模が比較的小さい場合などには直接営業部員が担当する。（勿論開発部門の協力は必要とするが）一方営業部門が独立していない場合や大規模プロジェクトの場合，開発部門の責任者やＳ．Ｅ．が担当することになり，生産性という観点から眺めると必ずしも得策でない。しかしこのどちらかの形態をとるか否かはプロジェクト規模，企業規模，組織環境や歴史といったことから決定せざるを得ない。

ただ発注段階に於いてどのような要員が担当するか否かは別問題として，大きな作業として図２－３の作業が考えられる。しかし発注先が一般企業か官公庁かに依ってはこの発注段階に於ける手続が異なる場合が多いので，ここでは一般的なモデルとして考察する。

図２－３　発注段階に於けるモデル




(7)　発注者側から眺めた場合

ソフトウェア・プロダクトに依っては開発当初から定量効果を明確に予測できる場合も多いが，しかし定性的にしか開発効果を評価し得ない場合もある，このような場合この効果を如何にアピールするかが発注者側の悩みとしてあるが，この評価問題に対する標準的な評価手法が確立されるとソフトウェア開発が更に促進されるものと思う。次に発注者側が持つ問題点を述べる。

①発注者側の意図が受注者側に十分伝わっているか，即ちコミュニケーション・ギャップがないか

②提出される見積がどの程度信頼できるのか，又この妥当性はどうか

③納品されてくるソフトウェア・プロダクトの品質問題や性能問題

はどうか

④ソフトウェア・プロダクトが納期通りに納入されるか

⑤ソフトウェアの生産に対して如何なる管理が行われるのか

(イ) 受注者側から眺めた場合

工業製品のように可視性のある，しかも誰れもが或る程度納得出来る物を作るのとは異なり，ユーザ・ニーズという目に見えない要求をこの段階ではどの程度迄明解にし，吸収することが出来るかで今後の作業に大きな影響を及ぼすことになる。

つまり受注者側はできるだけリスクをミニマイズする為に同種の開発を経験した要員の活用や開発経験の豊かな要員に見積をさせるなどして，かなり経験度を重視した活用を行っているが，しかしこの段階での見積技術がプロジェクトの採算や納期問題に大きな影響を及ぼしているのが実状である。この段階に於ける問題点を述べる。

①見積技術が個人的なknow－how として沈澱してしまい必ずしも組織として蓄積されにくい。

②見積技術が技術として必ずしも確立されていない。

③発注者側のニーズが必ずしも明確でない段階で発注者側の要請で非常にラフな予算を提出せざるを得ない場合が多く，しかもこのラフな予算が最終的な開発費用の足枷になってしまう場合が多い。

④システム開発技術がまだまだ未成熟で開発要員の個人的能力差に依って品質の良し悪しが決まる場合が多い。しかもどの機能もとり入れるとシステム規模にどのような影響を与えるかといった尺度がはっきりしないまま，発注者側の要望を取り込んだ為に採算問題や納期問題が発注する場合がある。

⑤契約内容に依っては，ドキュメント作成量といった工数に影響を及ぼすことがあるのでシステム範囲などと同様に十分検討する必要がある。

⑥システム範囲が必ずしも明確でなく，何に依ってシステム範囲を規定するか十分検討する必要がある。

以上のようにこの発注段階を受注者側，発注者側から眺めると様々な問題点がある。これらの問題点解決のために或る種のガイドラインとして標準化項目が設定されると発注者，受注者の双方にとって多くのメリットが期待できると考えられる。

(ウ) 標準化項目

ソフトウェアの特異な点として，ソフトウェア自体の優劣の判定が難かしくしかもシステム・エンジニアといわれる要員とその他に紙・鉛筆・コンピュータといった道具がありさえすれば開発することが出来るわけである。つまり標準化といった粋を遵守しなくてもプログラム言語の文法といった言語とのインタフェースさえマスターすれば開発する事が可能な訳であり，それと同時に大きな問題点でもある。手工業的な生産で我慢できる間はこれでも良いだろうが，最近のハードウェア・コストの低価格化傾向に対して，企業レベルでさえも開発・運用・保守といったソフトウェア・プロダクトに係る要員数の増大傾向があり，ソフトウェア，コスト問題や生産性向上問題が一層大きく浮かび上ってきた。まして国家的な見地からソフトウェア問題を眺めてみるとソフトウェアの流通や分業化のからみで早目に標準化の粋を設定しておいた方がベターと思われる項目について以下に記述する。但し企業レベルでも必ずしも十分標準化が浸透している訳でもなく，競争原理を阻害しない程度でしかも企業の独自性のもとで生産性の向上がはかれる必要があり，その意味で標準化の項目ならびに標準化の程度をどこに置くかが重要な課題となる。

① 要求仕様を表現する言語の標準化

ソフトウェアに要求される事項はシステム化ニーズに依って色々と異なっている為要求仕様という項目を標準化することはなかなか

困難があると考える。そこでその項目を標準化する代りに要求仕様を表現する言語を標準化することによって受注者側と発注者側の情報流通を円滑にする必要がある。このことは発注者側が持っている要求という情報を開発する立場にある受注者側に容易に伝えることが可能となり両者の大きな生産性向上が期待出来る。

その上要求仕様確定段階に於いて正しくない要求，一貫性のない要求，不明確な要求や抜けのある要求といった要求そのものの誤りを後続の設計段階や製造段階に持ち込むと誤りはどんどん増幅されてしまい，その誤りの発見が後続の段階になればなるだけ修正の為に大巾な修正作業を必要するので，どのようにして正確な要求仕様を確定するか又如何にして早い段階で要求仕様上のミスを発見するかが大きな課題となっている。しかし要求仕様化技術の現状を眺めてみると，実際に使用されたり研究中のものもあるがまだ必ずしも十分な評価を得ていないので今後とも研究を期待される分野である。この要求仕様言語が一般に利用される為には誰れもが簡単に理解出来るといった理解容易性，誰れもが簡単に作れるといった作り易さ，更に適用範囲の広さなどが必須条件となろう。

② フィジビリティ時点から発注時点迄の発注者・受注者間の情報報交換手順の標準化

この段階に於ける手順は各企業が長年培ってきたknow － how的な企業毎の独自性があり国家的なレベルでの情報交換手順の標準化が進めにくい分野である。

発注者側が持っているアプリケーション知識と受注者側の持っているアプリケーション知識のギャップを如何にして縮めるかが重要なポイントである。この為に受注者側は出来るだけ類似ソフトウェアの開発経験者などを割り当てることなどでこのギャップを解消する努力をはかっている。しかし常にこの条件にあてはまる要員を割

り当てられるとは限らないので効率良くしかも的を得た方法で発注者側の持っているアプリケーション知識を吸収する必要がある。この場合情報交換の手順の企業レベルでの標準化がなされているだけでもかなりの生産性向上に寄与するものと思われる。その上エンドユーザ自身が要求仕様を受注者側に提供出来る力を持つことにより一層の生産性向上につながると思われる。この場合エンドユーザの持つ要求仕様を導き出す為の技術であるシステム分析手順の標準化も重要な標準化項目である。

又アプリケーション・ギャップと同様に重要なことはコンピュータ知識ギャップ問題である。エンド・ユーザはコンピュータ知識について全然知らないことを口実に（実際にはかなりの知識を持っている。）過大要求を受注者側に押しつけてきたりする場合がある。この場合発注者側の要求が開発期間や費用といった面に直ちにはねかえせる見積技術があれば発注者側と受注者側とのバランスのとれた要求仕様にまとめる事が出来るので標準化項目の整備と同様にこのようなツールといったものの整備も必要である。

③　契約条項の標準化

Boelumは１９８５年に至る米国に於けるソフトウェア・コストとハードウェア・コストとの割合を９：１と予測しているようにソフトウェア・コストの重要性が今後も益々増大してくると思われる。ところが過去のソフトウェアに関連する取引はあくまでもハードウェアに付帯するものであるといった考え方が強かった。しかしコストに占めるソフトウェア・コストの増大に比例してハードウェアの一部といった考え方でなく立派な商品として取り扱われるようになってきた。しかしこの商品もして扱われるようになってきたソフトウェアに関する契約条項を眺めてみると，ハードウェアに関する契約条項が慎重かつ詳細にわたって規定されているのに反して，これだ

けソフトウェアの重要性が叫ばれている割にはあまり大きな注意が払われていないのが実状である。其の上開発されるソフトウェアは巨大化複雑化になれ，しかもソフトウェアが影響を及ぼす領域は拡大傾向にあるので契約問題は今後受注段階に於ける重要な作業となってくる。この意味で受注者側と発注者側の両者にとって契約条項の標準化をはかる事によって，プロジェクトの取引毎に作成する必要がなく契約作成作業の生産性に大きく貢献するだけでなく，事前に十分検討して契約条項を作成しておくので，漏れや検討不十分で発生するリスクを削減出来るといった法的安全性が高くなる。

更にソフトウェアの果たす機能が企業の根幹的なものや社会活動の重要な部分を占めるようになるとソフトウェアが引き起こす問題は非常に大きな問題となり契約条項如何によっては受注会社の存続さえも危くするほどの打撃を受けるので十分注意をする必要がある。しかしこのリスクを最小限にする為に受注者側だけに一方的に有利な契約条項では発注者側も納得しないので，両者の利害がバランス出来るような契約条項を検討する場合には次の諸点に十分配慮する必要がある。

①ソフトウェアの瑕疵に依って引き起こされる損害賠償をどの程度にするかが重要な条項となる。契約如何によっては膨大な損害賠償を請求される可能性があり，このリスクを回避する為にソフトウェア自体に防止の仕掛を求めると開発費用が逆に大きくなるので，両者にとってバランスのとれた合法的な金額を契約条項として締結する必要がある。

②ソフトウェアについて認められる権利問題について規定しておく必要がある。現段階では世界各国で様々な議論がなされているが，必ずしも統一のとれた見解が出されていないので，著作権や特許権に依ってソフトウェアが保護されるか否かの最終結論が出

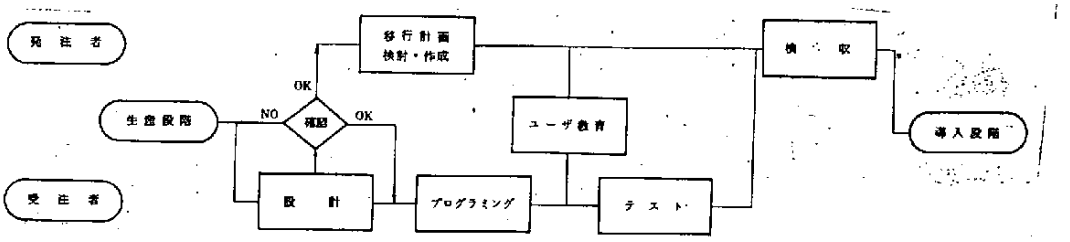
る迄はドキュメント類も含くんだソフトウェアの使用権の範囲を規定しておく必要がある。

③機密問題は発注者側の機密情報の取り扱いや受注者側の機密情報についても契約で十分取りきめておく必要がある。更にこの契約条項も契約形態（売買契約，賃貸借契約，請負契約，委任契約）に依って決定する必要がある。

(ii) 生産段階

発注段階で正式に認められ，しかも明確化された要求仕様に沿って実現していく段階が生産段階である。契約形態に依っては異なるが基本的には発注者側は，受注者の生産状況を見守っていく立場にある。一方受注者側は発注者側の持っているアプリケーション情報や運用環境情報などを最大限に吸収しながら約束の納期に間に合うようにプロジェクトの進行をはかる段階である。生産者に於ける発注者側と受注者側との作業分担を一般形として図２－４のモデル図として表示する。

図２－４　生産段階に於けるモデル




(1) 受注者側から眺めた場合

この段階は受注者側の生産方式や体制などに依っては生産性に大きな開きが発生する工程であり，標準化や工程管理が十分推進されている企業では納期問題はあまり発生しないが，そうでない場合はプロジェクトに携っている要員に依存する場合が多く，プロジェクトに割り

付ける要員を決定する場合には能力，経験，性格などを十分見定めた上で決定する必要がある。この段階に於いて受注者側が持つ問題点を下記する。

①ソフトウェアを生産する際の生産性向上をはかりにくい。

②システム設計上の品質がなかなか一定化しにくい。

③ソフトウェア生産の方法論が必ずしも統一化されていない。

④ソフトウェアの品質保証に係る費用を開発コストに反映しにくいのでどの程度迄をコストに反映させるかが問題。

⑤プログラミング作業の管理が困難

⑥プログラムの品質を一定化しにくい。

⑦ソフトウェアの生産規模が同じ程度であってもソフトウェアの持つ複雑性に依っても開発期間に大きな影響を与える場合などがあり，ソフトウェアの生産コストに深い関係がある要素（例えば，規模，複雑性，要員の能力，経験度などの）との関連が不明確。

⑧作成するドキュメンテーションが個々の契約に依って異なる為に，ドキュメント量と工数関係が不明確。

(2) 発注者側から眺めた場合

この段階に於ける発注者側の役割は，受注者側の進捗報告の信憑性を検査し問題の発生を未然に防止する必要がある。又新らしいソフトウェア・プロダクトをどの様にして現在の仕組の中に円滑に導入していくかの移行計画とシステムの対象となっている範囲の要員をどの様にして教育していくかの計画を立てて受注者側の協力のもとに実施を行う。この段階に於て受注者側が持つ問題点を下記する。

①進捗状況を受注者側から報告を受けるがこれがどの程度信頼出来るか

②運用マニュアルなどのドキュメントがどの程度整備されているか

③納品されるプログラムの保守性はどうか，簡単に修正が可能か

④移行上の問題はないか

(3) 標準化項目

この段階に於いて生産性向上や保守性向上などを目的に標準化を各企業とも積極的に推進しており，しかも重要性の認識は非常に高い。しかし現状を眺めてみると標準化の推進状況必ずしも満足のいく状態でない。この事は標準化の推進の困難性を端的に表わしている。そこで発注者側と受注者側の共通認識の為にソフトウェア生産の基本的な工程についての標準化をはかることに依り両者の情報流通を円滑に行うことが出来る。但し工程区分のレベルをあまり詳細なレベルに設定すると考え方の相異などで必ずしも統一がとれなくなるので大きなメッシュでとらえる必要がある。

① 生産工程の標準化

ソフトウェアという目に見えないものを管理する場合，その管理をどの様に行うか重要な課題である。この為にはソフトウェア生産の進捗を可視性のあるものにして検査が出来るようにする必要がある。つまりソフトウェアの生産段階を幾つかの工程に区分けして，分割された工程での作業が終了した事を示すドキュメンテーションを提出させる事に依って管理しようとする方法である。

しかし一方ではソフトウェアを生産する為の方法論が一般に市販されたり研究されたりしている。この場合，工程区分は必ずしも統一されておらず，しかもその工程につけられている名称さえもかなりばらばらである。ただ市販されている方法論のなかには工程区分の考え方そのものが一種の know－how であり，統一化出来にくい部分でもある。

但し今後もこの様な様々な考え方が出現し混乱を生じてくると思われるので，事前に大まかなレベルでの生産工程の標準化をはかることに依って発注者側と受注者側の意思疎通が円滑になり共通の土

台のもとでの討議がなされるので生産性向上に大きく貢献していくものと思われる。この工程の終了段階に作成するドキュメンテーションの標準化として1980年の4月16日のハーグでの会議に於いてDP6592 GUIDFLINE FOR THF DOCUMENTATION OF COMPUTER BASED APPLICATION SYSTEMS として提出され現在 ISO内で検討されている段階である。

⑪ 各工程毎の要求実現検査の標準化

工程の標準化で述べたように工程の終了は作成されたドキュメントを検査する事に依って判断することが出来るが，工程の標準化と同様に大切なことはドキュメントとして作成された内容をどのようにして検査するかである。しかしソフトウェアの特性に依っては記述する内容は千差万別となり，内容に立ち入っての検査はかなりの困難が伴うので，少くとも工程終了段階でのドキュメントにどのような項目を記載するかを標準化項目として設定しておけば，両者の検査作業を容易にする事が出来，しかも漏れといったことがなくなり品質向上にも大きく貢献するものと思われる。この事に関する標準化の動向として日本がISOのTC97／SC7 に提出しているプログラム・ドキュメンテーションなどがあり国機的な舞台でも積極的に取り組みつつある。

(iii) 導入段階

この段階はSEなどによって行われたインテグレーション・テストが終了し，現実の物にソフトウェアを埋め込む段階である。この段階での作業は主に発注者側のものであり，受注者側の作業としてはユーザ教育が重要な作業として発生する。この段階での作業を上手にやるか否かに依っては，折角作り上げたソフトウェアに対する詳価を一度に落とす事にもなりかねないので，導入に当ってはユーザと導入スケジュールや要員教育に事前に十分検討しておく必要がある。この場合，特に注意して

おかなければならない事は，ユーザ側にあまり耳慣れない専門用語を用いて拒絶反応を引き起こさせないように十分配慮する必要がある。この段階に於いて標準化項目として検討すべき項目として次の事項が考えられる。

① 導入手順の標準化

⑪ 検査方法の標準化

⑪ 用語の標準化

C. 用語

インタフェースの場では，適確な情報の伝達が最も重要なことであるので，用語とその定義の標準化が必要であることは，いうまでもないことである。どのような事項に関する用語の標準化が必要であるかは，既に上述の各所で述べてきたので，ここでは，特にソフトウェアに関する用語で，非常に困難ではあるが是非標準化すべきものについて改めて指摘する。

それは，処理機能の標準化である。2.1において，今日考えられる各種のソフトウェアが一応体系化されているので，それぞれの適用分野ごとに，標準的と考えられる処理機能又は機能単位をえらび，その内容の定義とそれの名前とを規格として定めることである。

標準的処理機能は，いくつかのレベルに互ってえらぶべきであろう。詳細なレベルでは，分野別はあまり意味をもたなくなり，むしろプログラムの基本構造の機能の選定になる。

この標準化が行われると，ソフトウェアの設計・製造の場及びインタフェースの場において，したがってまた利用の場において得られる利益は極めて大きいことは，上述のことから明かであろう。

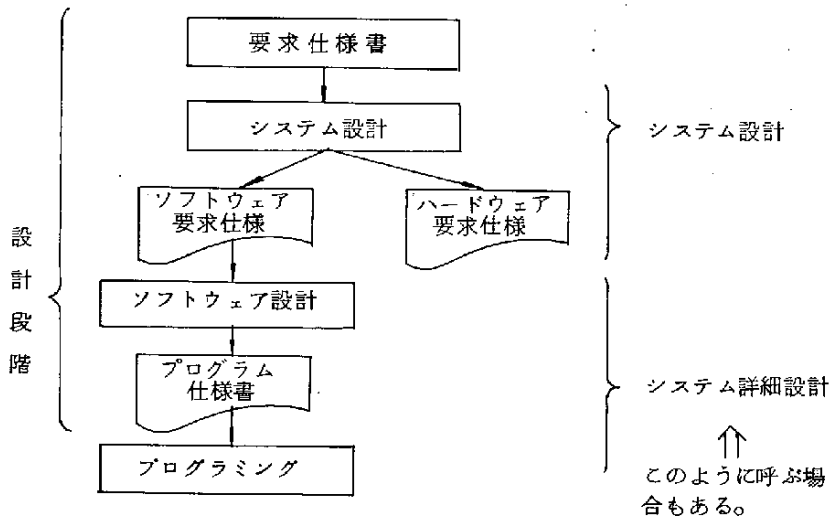
(4) 効率よく生産・利用するには

A 設計面

設計段階は要求仕様段階で得られた「システムまたはソフトウェアが何を目的としているのか，そして何をするものであるか」を記述したドキュメントである「システム要求仕様書（要求を定義したもの）」を入力として次の様な作業を行うものである。すなわち，設計段階は，ユーザの要求に適合したシステム（ソフトウェア）をどのようにして実現して行くかを検討する。この結果としての出力はシステム設計仕様書である。（一般にソフトウェアまたはプログラムの要求仕様と，それを実行すべきハードウェア要求仕様から構成される。）

必要なソフトウェアを実現するためには，設計段階としてさらに，ソフトウェア設計を実施することが必要となる。

ソフトウェア設計の段階では，ソフトウェアに要求される仕様を実現するために，それが使用するデータ構造とプログラム構造を決定し，プログラムとして実現することを可能にする。ソフトウェア設計では，できるだけ容易にプログラミングできる様に（あるいは容易にプログラミング出来る程度の大きさと，複雑さをもつ様に）プログラムを分割しておくことが必要となる。



プログラム設計の工程は次の二つに大きく分けることが出来る。すなわち，システム性能（応答時間，処理しなければならない最大負荷，必要なメモリの量など）を満足する処理方式を決め，入出力データと重要な内部データ構造として，モジュール実行を制御するための制御表，ファイル構造などを決める。

一般にプログラムが大きな場合には複数の人がチームとして担当するので，設計の規約などの標準を定めておくことが必要になる。

この作業につづき，条件にあったプログラム・モジュールを切り出し，モジュール間のインタフェースとモジュールの外部仕様を決定して行く。このようにして作られたモジュールの集合がソフトウェア要求仕様を満足しているかどうかを検討チェックするために，設計のレビューを行うことが必要である。この結果によって設計が承認されることになる。

設計段階はソフトウェアのライフサイクル全体からみて，後工程にあたえる影響も非常に大きい位置をしめている。したがって，設計フェーズに十分力を入れることが，品質もよく，構造のよいソフトウェアの作成にとって重要なことが立証されて来ている。また，設計工程は要求仕様とともに最も人間自身の活躍する段階である。

このため，各ステップでの作業結果，（各ステップで行われた作業の出力として次のステップに入力として使用される文書）が，わかりやすいものでないと，次工程の作業工数だけでなく，品質にも大きく影響する。したがって，設計段階各作業の標準化と，その出力である文書類（ドキュメント）の標準化が非常に大きな効果をもつことが明らかである。

最近はこのような面から，作業標準，標準文書形式（フォーマット），作成基準などまで，規定することによって，なるべく均質な水準をもつソフトウェアが生産できる様にするための設計標準を利用する傾向が非常に進んできている。ＳＴＥＰＳなどはその代表的なものといえよう。

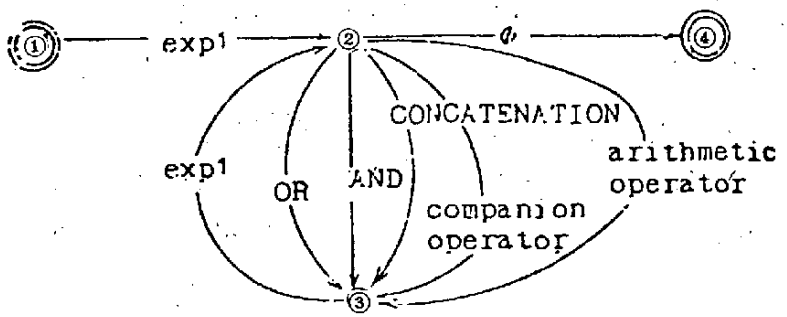
次の問題として，ソフトウェアの生産性と品質の向上は現在ソフトウェ

アがおかれている状況から最も重要な問題点となっている。設計作業の効率化もこの面からの考慮が非常に大きなポイントである。ソフトウェアで要求仕様を実現する方法の中で，機能の標準化が適切に行われれば，設計工数及びその後のプログラミング段階での効率向上に大きな効果がある。実際，ＳＴＥＰＳの考え方の基本としてのソフトウェアの標準機能を設定し，その組合せとして個別ユーザプログラムを設計する方法（見方をかえると，ソフトウェアの部品化設計方式ということが出来る。）が大きな成果をあげている。

機能の標準の例と，設計段階における標準ドキュメントの例をあげると表Ａ－１のようになる。

表現法についてみると，ソフトウェア設計工程を，少しでも人間にわかりやすい形で表現できることは効率化と品質の向上の面からもかなり大きな意味をもっている。

これらの例として，表形式による表現法，状態遷移図，ペトリネット，タイムチャート，ＮＳチャート，ＨＩＰＯ，ＰＡＤなど，最近多くのものが使われる様になっており，それぞれの適応領域を十分考慮した上で利用することによって，かなりの効果があると思われる。この様な表現法は，システム設計の段階の表現に適したものが多い。標準化の面からこれらの機能，利用可能範囲なども含めて検討が必要になるだろう。これらの方法について共通な特長はいづれも２次元的な表現法であり，平面上に図形として表現することが，人間の思考の方向との関連性が強いとも考えられ，これらをうまく利用することで，設計の品質面の向上も十分期待できよう。




状態遷移図の例

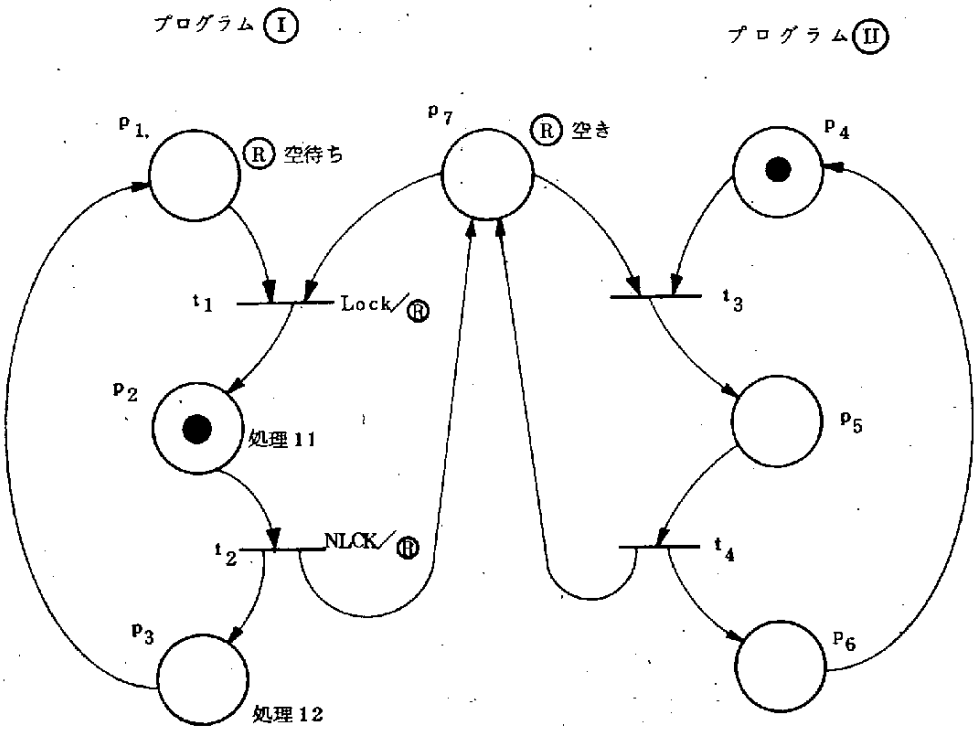
表A－1　ドキュメント一覧

| | システム分析 | システム概要設計 | システム詳細設計 | 製造 | 導入準備 |
|---|---|---|---|---|---|
| 仕様書 | システム概要仕様書 | システム仕様書（編構成）<br>・ジョブ仕様書<br>・業務仕様書<br>・入出力仕様書<br>・画面仕様書<br>・コード仕様書<br>・JCL仕様書<br>・コマンド／メッセージ仕様書<br>・マクロ／ライブラリ仕様書<br>・テスト仕様書<br>・移行仕様書<br>・DB(ファイル)仕様書 | プログラム仕様書<br>モジュール仕様書<br>画面製造仕様書<br>JCL製造仕様書<br>マクロ/ライブラリ製造仕様書<br>テスト実施仕様書<br>移行実施仕様書<br>DB(ファイル)製造仕様書 | | |
| 分析書<br>設計書 | システム分析書 | | コマンド/メッセージ設計書<br>入出力設計書 | | |
| 説明書 | | | コード説明書 | プログラム保守説明書<br>システム運用説明書 | |
| | | | 業務運用説明書 | | |
| 計画書<br>報告書 | システム開発計画書 | | | テスト完了報告書 | |
| | 技術評価報告書 | | | | プロジェクト完了報告書 |
| 標準 | 技術標準 | | | | |

備考　フェーズ間にまたがっているドキュメントは，その間で作成するか，あるいは最終的に完成すればよいことを示す．

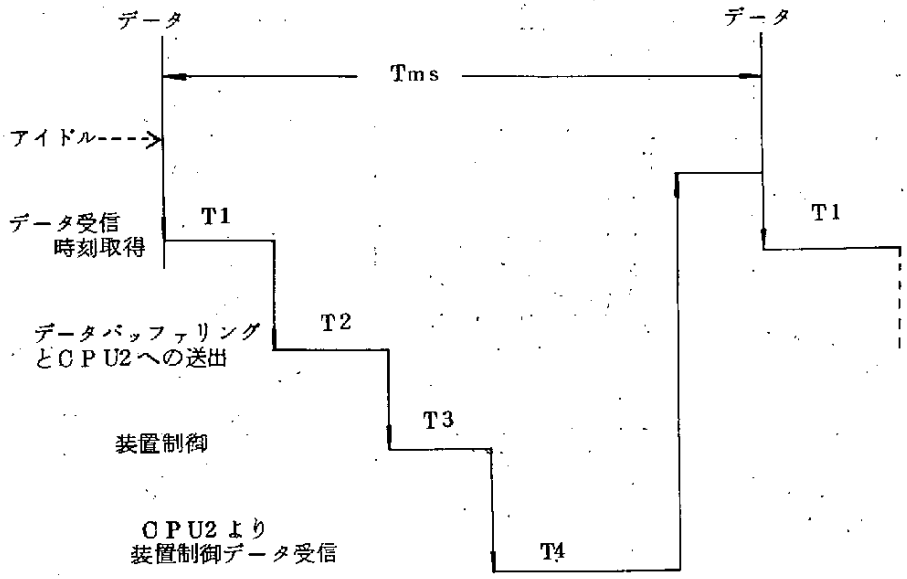
表A－2　ペトリネットの記号

| 項番 | 記　　号 | 名　称 |
|---|---|---|
| 1 | Ⓟi | 場　所<br>（place） |
| 2 | ——— t i | 遷　移<br>（tramsition） |
| 3 | ——→ | 有向枝 |
| 4 | ● | 石またはマーク<br>（token） |

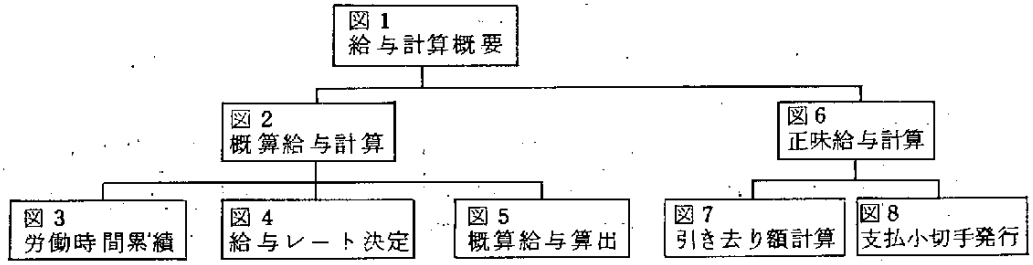



ペトリネットの例

| 番号 | 名　　称 | NSチャート記号 | フローチャート記号 |
|---|---|---|---|
| 1 | 順次処理 | 処理1<br>処理2 | 処理1<br>処理2 |
| 2 | 2分岐選択<br>(IF) | 条件<br>T　F<br>処理1　処理2 | 条件　ELSE<br>THEN<br>処理1　処理2 |
| 3 | 多分岐選択<br>(CASE) | CASE<br>条件1　条件2　OTHER<br>処理1　処理2　≈　処理0 | CASE<br>条件1　条件2　OTHER<br>処理1　処理2　処理0 |
| 4 | 繰り返し<br>(WHILE) | 条件<br>処理 | 条件　ELSE<br>THEN<br>処理 |
| 5 | 繰り返し<br>(REPEAT) | 処理<br>条件 | 処理<br>ELSE<br>条件<br>THEN |
| 6 | 呼び出し<br>(CALL) | 処理名　EXT | 処理名 |
| 7 | 並列処理 | 処理1　処理2 | 処理1　処理2 |




タイムチャートの例（２CPU上のある制御機能のCPU１のタイムチャート）




| | |
|---|---|
| 1 | 給与管理システム中の給与計算の概要 |
| 2 | すべての概算給与計算機能の概要 |
| 3－5 | 概算給与計算の設計の詳細<br>3－労働時間累積<br>4－給与レートの決定<br>5－概算給与の算出 |
| 6 | 正味給与計算の概要 |
| 7－8 | 正味給与計算の詳細<br>7－全ての引き去り計算の実行<br>8－給与支払小切手と明細書の作成 |

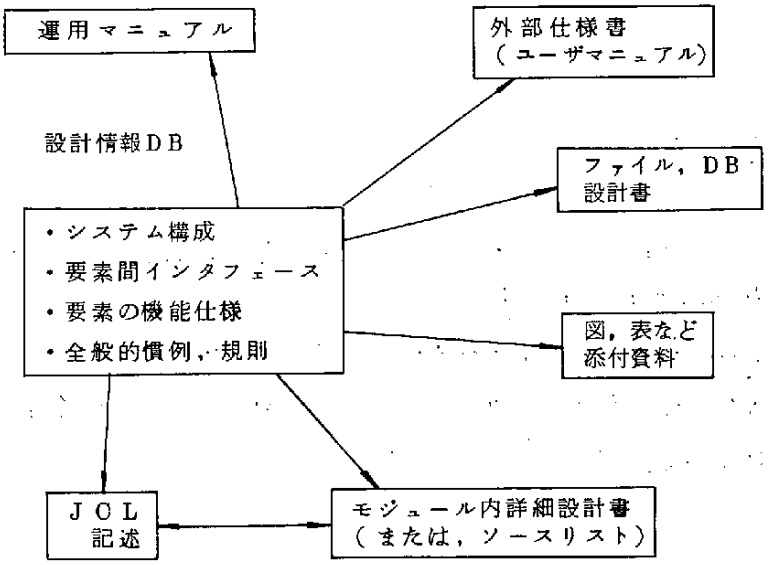
ＨＩＰＯの例：給与計算階層図

## A1　設計情報ＩＲ

設計情報は先きにのべたように多くの文書類から構成される。従来これらの文書（ドキュメント類）は一般に手作業で行われて来ており，表現法も自然言語の利用部分が大部分をしめて来た。しかしながら，システム毎に多くのソフトウェアが開発され，また保守されていくソフトウェア量の増大を考えると，コンピュータのソフトウェアを作るのに，コンピュータを一層利用するという考え方が強くなって来ている。このため，ソフトウェア設計情報のＤＢ化を行い，その検査，更新追加などが必要な時点，場所で出来る様にする試みがなされて来ている。

設計情報の整理と文書情報としてのデータ構造，文書表現方式などの検討が必要であろう。ＤＢを基礎にして，各種の設計情報文書をある程度，コンピュータで自動的に作成すること，また設計情報の相互間のクロス・レファレンス表なども要求されるであろう。

設計情報ＩＲシステム



## A 2　設計原価の計算

ソフトウェア作業のコスト見積り，工数見積，規模期間見積りなどは性能予測と同様にユーザ，作成側いづれの側にとっても大変重要な意味をもっている。コスト，工数，規模，期間の見積りとそれを開発実行中または保守実行中を含めて各時点時点で見直しを行えば，実績との差異をみてより適切な管理ができるだけでなく，このような見積り，見直し，評価を繰返すことによって，ソフトウェアの開発――設計段階で見ても，その開発の方式，内容，などについて，大局的な見地から生産性，品質の面からの向上等なども検討することが出来ると思われる。

従って，コスト見積り，コストモデルの標準方式の設定は大変重要なことと考えられる。これらの方式については，最近米国のTRWのBoehm，DODのDotyなどの研究が発表されており，また，Rayleigh Curveとソフトウェア開発保守工数との関係についての研究などがある。また，より現場的な方法としては，まずデータを収集して，各開発projectにおける設計段階の工数，費用，さらに設計対象の領域別，機能別のデータ蓄積から予測する方式の開発が望まれている。この際これらの方法はできるだけ使いやすいものでなければならないことは勿論であろう。

Boehmのモデル

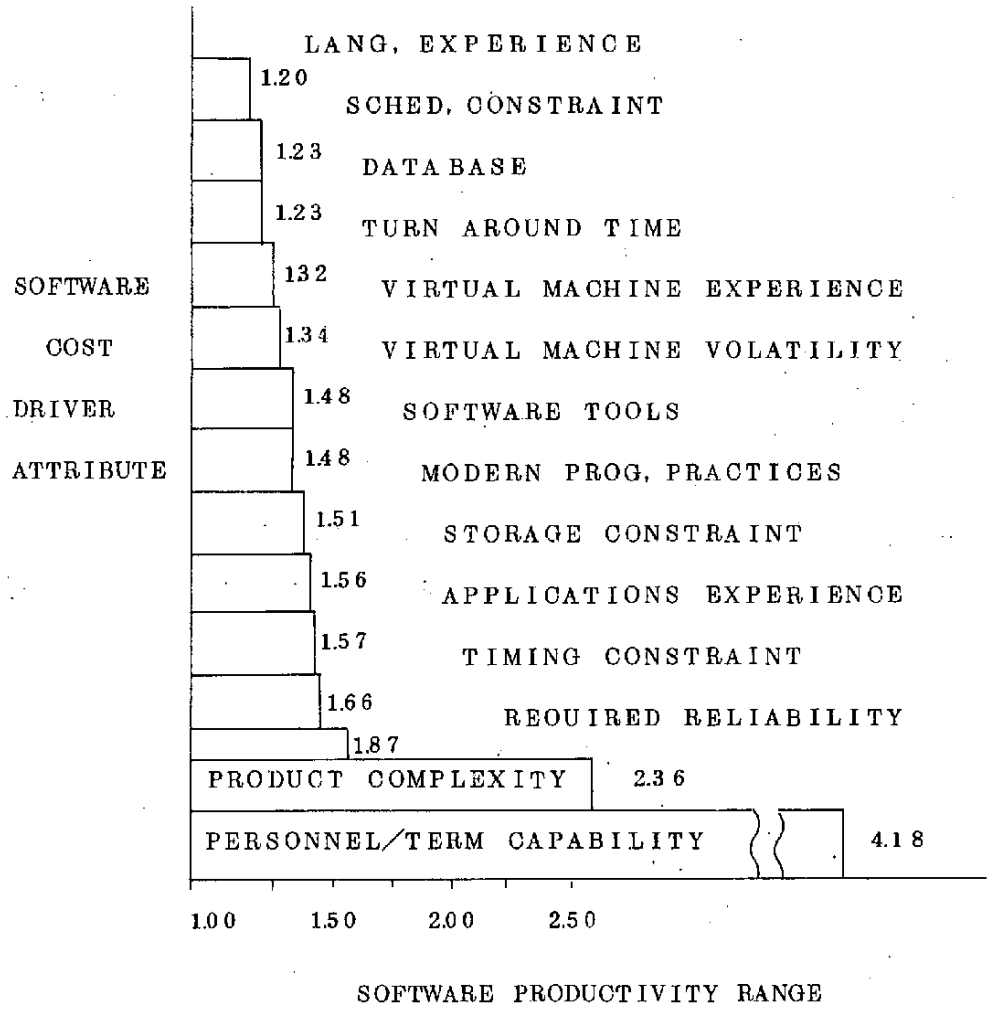
$$MM = 3.5\ (KDSI)^{1.05}$$

KDSIはソースプログラムステップ数（Kilostep単位）

MMはman－month

名目工数でこれに14種類のパラメータを掛ける。

大型で複雑なソフトウェアの場合は指数がもっと大きくなるといわれている。




## B　製造面

### B.1　ソフトウエア製造における標準化

ソフトウェア・エンジニアリングは着実な発展が行われ，多くのソフトウェア開発の場で生産性や品質を向上させるための開発規約等が作られ効果をあげてきている。

しかし，ソフトウェアは大規模化し，多様化するため，一つの部門だけで

開発を行うことが困難となり，共同開発や外注が一般的になってきている。

このような場合，工程のたて方，工程の管理，ドキュメントの作り方が標準化されていれば，工数見積りも確かになり，工程もスムーズに進めることができる。

標準化された工程や工程管理を経て開発されたソフトウェアは，或る程度の信頼性を保つことができ利用者にとっても安心できる。これはソフトウェアの外注や流通を更に拡大するためにも重要なことである。

更に，標準化された工程や作業内容に従ったソフトウェアについては生産性や品質について共通の土俵に立って評価ができるため，ここから生まれる生産性や品質の向上策は多くの場に具体的に適用できる。

このような観点からソフトウェア製造面において最低限必要と思われる標準化の項目について述べる。

### B.1.2　ソフトウェア製造における工程とドキュメント

製造段階における各工程とその名称を決め，その工程でどのような作業を行うかを決めておく必要がある。

できれば，各工程の定義或いは作業内容は厳密に決めなくともその工程の生産物であるドキュメントを内容も含めて明確に規定することにより，工程の定義となり得ると思われる。

製造段階の工程も，ソフトウェアの規模や用途によって，一つの工程を更に分割した小工程を設けたり，ソフトウェアの導入と運用を重視した工程を加える場合もある。

しかし，製造面における工程の基本は同じなので，大まかな工程でも，より細分化された工程でも，どちらにでも対応できる工程の標準化を考える必要がある。

これら工程とその過程で作成されるドキュメントの標準化ができればソフトウェア製造における標準化の大きな部分が完成したことになると思われる。

以下に標準化の参考として，製造の工程とドキュメントについて述べる。

まず製造とはソフトウェア開発のどの工程からを言うのかが問題になるが，ここでは，プログラムがこれを構成する最小の翻訳単位（コンパイル／アセンブル単位）であるモジュールにまで分割されているとする。更に，各モジュールの機能やモジュールに対する入出力，各モジュールが使用する共通テーブルや作業用ファイルなども決定されているとして，このモジュールの機能を処理手順として詳細化する工程からを，製造と考える。

従って，製造での工程としては，

- モジュールの詳細設計
- モジュールのプログラミング
- モジュールのテスト
- モジュールを結合したテスト

が考えられる。モジュールの詳細設計は製造工程ではなく設計工程としてもよいが，コーディングに近い詳細化という意味で製造として考える。

(1) モジュールの詳細設計工程

(i) 作業内容

モジュールの機能に従って，コーディングが可能な程度に処理手順を詳細化する。

(ii) ドキュメント

- 詳細仕様書

主な内容としてはモジュールの処理手順やモジュール内部で使用するテーブルやデータ定義を述べる。処理手順は文章で記述したり，ＨＩＰＯや擬似コードを用いてもよい。

(2) モジュールのプログラミング工程

(i) 作業内容

詳細仕様書に基づきモジュールのコーディングを行い，アセンブルまたはコンパイルしたあとモジュールのテストを行う。

(ii) ドキュメント

- コンパイルリストまたはアセンブルリスト
- モジュールテスト仕様書

  テスト項目毎にテストの目的と具体的な入力データ及び出力結果を述べたものである。
- モジュールテスト成績書

上記のテスト仕様書に基づいてテストを行った結果を述べる。また，障害種別毎の発生件数，障害作込工程別の発生件数などの品質データをまとめると同時にモジュールの品質評価や今後の障害発生予測等を述べる。

なお，この工程でモジュールのステップ数やメモリサイズがほぼ確定するのでこれらについても今後の基礎データとして収録しておく。

(3) 結合テスト工程

(i) 作業内容

モジュールを結合しながらモジュール間及びプログラム間のインタフェースの正当性をテストする。

更に，全体としての機能，性能及び操作性について目的に合っているかをテストする。

(ii) ドキュメント

- 結合テスト仕様書

  モジュールが結合されテストする順序とテスト項目毎のテスト目的及び具体的な入力データと出力結果を述べる必要がある。
- 結合テスト成績書

  結合テスト仕様書に述べるテスト項目に対するテスト実施結果を述べる。

  また，モジュールテスト成績書で述べたような品質データのまとめも必要である。性能については別項を設けて部分的ばかりでなく総合的な評価が述べられるべきである。

### B.1. 2. ソフトウエア製造における管理

ソフトウェア製造における管理は職場の組織，歴史や経験により様々で管理の標準化は困難と思われるが，進捗管理，予実管理及び品質管理について標準化が必要と思われる項目について述べる。

(1) 進捗管理

- 工程線表の書き方
- 工程線表の管理・更新の手続き
- 工程会議の開催と進捗状況の報告単位及び報告内容

報告内容については作成済ステップ数，作成済テスト項目数，作成済ドキュメントのページ数，テスト済のテスト項目数及び発見したバグ件数などは少なくとも報告内容に入れることを標準とする。

- 各種指標の分析

(a) プログラムの生産性＝ステップ数／人月

(b) ドキュメントの生産性＝ページ数／人月

(c) テスト項目設定率＝テスト項目／ステップ数

などは標準化項目として取入れたい。

- ドキュメントの管理と変更手続き

（個人が勝手に修正したり持出したりしないため）

- ソースプログラムの管理と変更手続き

(2) 予算実績管理

- 工程別工数（又は費用）の予定と実績
- 工程別計算機使用時間（又は費用）の予定と実績
- 消耗品等の費用の予算と実績

などは少なくとも標準化の中に取入れたい。

(3) 品質管理

- バグの定義

どのようなバグをバグとして取上げるか。例えば，ドキュメントの誤字

脱字はバグとして取あげるかとか，プログラムのバグはコーディングが完了しアセンブル／コンパイルして文法的なエラーがなくなってからカウントするかなど明確にしておく必要がある。

- バグの種別

設計のミスか，単なるコーディングのミスか，ドキュメントのミスかなどの分類

- モジュール毎のバグ検出目標と実績及び実績が目標に較べて多過ぎたり少な過ぎる限界を設定すべきこと。これに対しては再度モジュールのレビューを行うなどの対策を取るべきこと。

- レビューの明示

近年は各工程でのレビューの実施が一般に行われるようになり品質面で大きな効果を上げているため，各工程でのレビューを義務化するのも良いと思われる。工程の一部としてレビュー工程として取入れるのも一方法である。

- バグ票

全てのバグはバグ票に記録することにし，バグの集計や分析に役立てる。

バグ票の内容としてはバグの存在したモジュール名，バグの内容，バグの種別，バグを組込んだ工程などは記載項目として必要である。

- 品質の指標

(a) テスト項目設定率＝テスト項目数／百ステップ

(b) バグ検出率＝テスト項目数／バグ

更に，テスト項目に基づくテスト以外にテスト実施中に不都合が発見される場合がある。意識してバグを発見する場合とそうでなく発見された場合の比率を求めておくのもテスト項目設定の評価の一つとなる。

(c) バグ発生率＝バグ件数／Kステップ

ドキュメントについては

＝バグ件数／ページ

などを品質の指標として決めておけばよい。

## B.2　ソフトウエア部品について

### B.2.1　ソフトウエア部品の必要性について

ソフトウェアの生産性を高め，品質を向上させるために近年プログラミング方法論の発展があった。一方，ソフトウェア開発費の低減のため既存ソフトウェアの再利用が従来から行われている。

既存ソフトウェアの再利用は，ソフトウェアパッケージの形をとるのと，既存ソフトウェアを異なるＯＳやハードウェア上で動作させるためにコンバータを利用するという形が一般的である。

しかし，現実にソフトウェアの開発を行う場合にも既存ソフトウェアの再利用を行うことが考えられる。

長い歴史を持つ工業製品の場合は，規格化された部品が存在し，当然の事として広範に安心して利用されている。ソフトウェアの製造においても規格化された部品を作り，この部品が活用されれば生産性と品質の向上を図ることができる。

ソフトウェアは適切なアルゴリズムの採用とプログラミング方法論にのっとったものであることが必要であり，設計者，プログラマ及びレビュー者がこれらのノウハウを熟知しいることが望ましい。

しかし，このような質の高いソフトウェア開発要員を確保することは現実には困難である。

この困難を少しでも軽減するには，アルゴリズムとプログラミングが十分に吟味されたソフトウェア部品を作成し，これを活用することである。

更に，ソフトウェア開発者の初期教育においても，良いアルゴリズムと良いプログラミングを豊富なソフトウェア部品から学ぶことができれば，ソフトウェア部品の効果は大きいと言える。

部品の定義は簡単ではないが，大きなものより，広範囲に利用され，利

用頻度の高い小さな機能，例えば特定の文字列を見つけたり，日付の算出など機能とする。

また，規模も数行から数百行のものに限るとよい。

### B.2.2. ソフトウェア部品化を進めるための標準化について

ソフトウェア部品化を進めるためにどのような標準化を考えておくべきかを以下に述べる。

(1) 部品の定義

- 部品の機能の範囲
- 部品の規模
- 部品が使用する言語
- ＯＳやハードウェアに対する部品の汎用性もしくは局所性

など部品として認める範ちゅうを明確にしておかねばならない。

部品の範ちゅうは一度決めたら変らないものではなく部品の普及や利用者の要望等により拡大し，場合によれば一部縮少されることもある。

(2) 部品のコード体系

部品の作成，集収，管理及び利用を容易にするため，部品のコード体系を整備しなければならない。部品の持つコードは必ずしも一つだけではなく複数個でもよい。コードは利用者が部品の検索の際に便利な体系になっているべきで，機能別コード，取扱うデータ別コードや使用言語別コードなどが考えられる。

このコード体系を標準化しておけばソフトウェア部品の流通も促進される。

(3) 部品仕様書

ソフトウェア部品を広範囲に利用してもらうためには利用者の必要に合った部品を正確に選んでもらい，正しく使用してもらう必要がある。

このためには，ソフトウェア部品の仕様書の記載項目と内容の書き方を統一しておく必要がある。考えられる項目としては，

・部品名と部品コード

・機能

・部品の使用法と使用上の注意事項

・部品の形態（ソースコードかロードモジュールか）と組込方法（マクロなど）

・使用言語

・ステップ数

・占有メモリ

・性能

などがある。

(4) 部品登録と管理

部品登録や管理のための書類の様式や手続きを標準化し，部品の利用や交流にそなえてずさんな管理が行われないよう標準化しておく必要がある。

(5) 部品の品質保証

部品作成におけるテストの内容や結果について，必要最低限のことが実施されていることが確認できる品質確認書を標準化して，部品の登録に際して品質の確認ができるようにすべきである。

(6) 部品のトラブル

ソフトウェア部品のトラブル（障害）は利用促進の大きな妨げになるので，障害発生時の報告，それに対する回答，及び障害修正とそれに伴なう部品の履歴管理などの手続きを標準化しておく必要がある。

(7) 部品の利用状況

ソフトウェア部品の利用状況は部品の開発，改良，削除や利用促進のための重要なデータとなる。

従って，利用状況を定期的に記録報告する様式を決めておくことが望ましい。

ソフトウェア部品の蓄積と利用が進むことはプログラムの製造だけでなく，設計も部品利用を前提として行われることになる。

従って，従来から言われたようにソフトウェアは一品生産という考えから，一品生産であっても多くの共通部分が部品というレベルで標準化される。

更に，複数の部品の組合せからより大きな機能を持つ部品ができるようになれば，夢と思われたソフトウェアの標準化に一歩近づくことになると思われる。

## C　ドキュメントの整備

### C 1.　ドキュメント整備の必要性

産業界において製品に付随するドキュメントの重要性は，しばしば無視されている。一見してその操作や特質が判るような製品は別として，現品を受取っただけでは，手のつけようもない複雑な製品は（それがハードウェアであれ，ソフトウェアであれ），詳細な説明書なしでは，全く意味がない。

いかに良い製品でも，その利用者は製造者ではない他人であるから，文書または口頭の説明を聞き（読み），ある程度の時間をかけて了解して，始めて利用し得る状態に達する。その利用の方法を誤れば，いかに良い製品でも，利用者にとって何の役にもたたないばかりか，更に損害をも引き起す。

従って製品に関し，その使用法の訓練を，製品販売上の必須の要件とすることが多い。現在ソフトウェアについて，製品販売と訓練は不可分のものと考えられるようになっている。そして訓練には，口頭だけではなく，書き物（普通は説明書）を提供することが，訓練の不可欠な部分となっている。

それにも拘らずこの説明書は，形だけは整っていても，内容が極めて不完全で，ユーザが不満を洩らすのが常である。不完全な説明書を添えた製品を売ることは，性能不完全な製品を売ることに等しい。このことは一応理解されてはいるが，現実には徹底していない。その理由のひとつは，製品がハードウェアであるときには，われわれが知らず知らずのうちに，ハードウェア優先の思想に支配されているため，ハードウェアが良ければ，他の面が不満

でも止むを得ない，と考える習慣があるからであろう。

一方製品がソフトウェアであれば，ソフトウェアが不備であれば致命的であるはずなのに，事実は不備なものが，まかり通っているのは何故であろうか。これは，売り手・買い手ともに，外形（すなわちソフトウェアのハード的な面）だけに拘って，中味は不問に付する傾向が強い。たとえば説明書を付けるという契約事項があれば，説明書がとにかく，あるかないか，少し進んでも何ページあるか，更に進んでも文字がきれいか，印刷が一応の水準にあるかを検査するに止まり，それ以上のことは行われない。

個々の契約に基く検査では，中味にまで立ち入っていては，検査自身が実施できない，という意見もある。そうであるなら，説明書に対する一般的な基準が存在し，それが高い確率で守られている。ときどきは抜取検査が行われている。とでもいう保証がなければ，買い手は安心できないはずである。

現状は，守るべき基準もない。基準に達しているか，調べる方法もハッキリしていない，従って守る熱意も生れない。かりに熱意はあってもこの基準を守るためのコストが認められていない，等々の理由があろう。これは，もはや個別問題では解決せず，全国的な大きな枠の中で解決法を求めるほかない。従って業界全体の問題として取り上げるほかない。

### C 2. ドキュメントと中味に関する特質

(i) 多量であること

ドキュメントというものは，本質的に量が多い。しかも，全ての関係者がすべてを読むというわけではない。即ち多くの人は，必要な都度，必要と思うところを読み，時間に追われるから関係個所を抜けなく完全に読むことはない，と覚悟しなければならない。

内容が極めて論理的で（ソフトウェアでは多くの場合そうである）あれば，関係するところを限なく読まないと欠陥，あるいは間違いが生じる危険があ

る。だから利用者は洩れなく読まなければいけない，と執筆者側が主張しても，これはマシンに関しては適合しても，通常の人間に対しては適合しない。マシンの場合には，冗長性のある論述を行うことは，むしろ間違いを引起す可能性があるので，冗長性がなく，形式論理的に，必要かつ十分なようにのみ記載すべきだと考えられる。しかし，これは人間に対して，特に学習過程においては，適用できない。

また憶えておくべき事柄の量，マシンでいえば外部記憶の容量に当るもの，はマシンの場合と大きな違いがある。つまり記憶量には制限があり，簡単に拡張できない。従って多量のドキュメントに対応するには，無条件ではダメで特別の配慮がいる。

また記憶の形成は機械ほど早くなく，一度記憶せよと内容を送り込んでも無条件で受け付けるわけではない。何回かの繰返しがあると良く記憶でき，しかもその繰返しのしかたにも何かある秘訣があり，単に同じことを繰返すより，異なる条件のもと，あるいは異なる他の事項との組合せで繰返すとよい結果が得られる。

しかし他方で「処理能力」では，人間の能力はマシンにくらべ極めて高い能力を示す。特に複数の（しかし余り多数でない）要素の組合せで行う判断——いわばリレーショナル的な処理——で特に著るしい。

従ってドキュメントの多量性に関しても，同一の判断を多数回繰返すような操作は人間の能力に依存せず，補助手段に任す方がよい。例えば情報処理機械を使わずとも，印刷物において，表の活用，索引の機械的な利用によるよう，説明あるいは記載を工夫すべきである。

ドキュメントの多量性を考慮してドキュメント自身の構成・構造をきめてゆかなければならない。

(2) 多様性

情報はもともと多様なものであり，それを記録し運ぶドキュメントも（内容が）多様になるのは当然のことである。無限に多様なものを単に併列的に

扱って，多様性を個別に提供しても，読者である人間はそれら相互間の差異を認識しにくい。

しかし多様なものも，類似なものは集めて一群（class）とし，さらにこの群を複数まとめて，より大きな一群（高次群，高位のelass）を作り，という分類方式をとると，人間の行う理解は顕著に高まる。

従ってある技術内容を記述し，読者の理解に訴えるには，そのものだけのことを述べるより，他のものとの類似，あるいは差異に関する説明を付加しながら行うと効果が大きい。

この点はソフトウェアの説明書では，必ずしも実行されていない。恐らく指導員が訓練するときには，この点を口頭で付加して行うが，テキストにはそのような配慮は（テキストを簡単にするため）必要でないと考えていることが多い。しかしこれは再考すべきであろう。説明書は仕様書ではないのである。

## C 3. ドキュメント記述の標準化

### C 3. 1. 記述の特性

一つのドキュメントは，取扱う対象が違っていても，おのずから一つの様式に従うものである。これには，論述というものには，およそ定った形式があるという事実と，一方このこととは別に，説明書とくにソフトウェアの説明書が持つ特性のために考えなければならない文書作成及び刊行上の特性とがある。

前者はソフトウェアには少しも限定されない，論述一般の問題であり，他人にものを説明するために備えるべき特性である。

これに反して後者は，ソフトウェアの現在の形態や特質から来るものである。例えば，頻繁に改訂があるとか，他の説明書の改訂により影響を受け，また他の説明書の内容にも影響を与えることが多い。つまり全体として一つのシステムを形成している，とかいうようなことである。

前者を一般形式，後者を固有形式とここでは呼んで区別することができる。

一般形式は技術的な記述文書に共通したことで，ここで詳細を検討することは避け，主要な事項として以下の点をあげるに止める。第一は論述の骨組の十分な検討（これが欠けると執筆者の頭の中にある主張・提言・思考のクセが直接出てきて，条件の違う他人には理解しにくくなる），第二に説明の必要要素に抜けがある（自身では当然とすることも，他人の眼から見れば思考に飛躍があると容易に感じられる）。第三に要素は全部揃っていても，要素の結び付けかたに欠陥がある（例えば直線的論理ですむ場合は簡単であるが，連立方程式的やループ的な結び付きのあるケースに対して説明ができていない。複雑な問題にはこの種の説明が必要なことが多い）。

「形式」以外で重要なことは文章作成能力である。この能力は今日の日本の学校教育の全課程を通じて不足しているので，積極的な訓練が必要である。ことにソフトウェア関係の若い従事者には基本的な訓練（本来ならば初級学校教育で身に付けているべきことで，業界としては放念してよいはずの訓練）が改めて必要である。

その訓練の多くは講義だけでは大した役に立たず，実施訓練ないしは演習ではじめて体得できるものなので，訓練時間と指導者のマンアワーを費すことが多い。ただし学習者に積極的な意欲があれば，自己訓練でも相当の成果が得られることなので，職場の周囲の者や上司が指導するならば，入社後の数年間を通じて向上しうるものである。従って職場全体が，ごく初歩的な文字の書き方から始め，正確な文章を書くことに気を使う習慣が確立すれば解決することなのである。その習慣が容易に育成できない場合には，経営側が相当の負担をして訓練を行わなければならない。

C 3. 2.　固有形式

第二の固有形式については，問題が多い。

(1)　読者のレベルの設定

説明者は読者がいかなる人びとであるかを明瞭に意識しなければよいものは作れない。しかし現実に読者のレベル（主に豫備知識）はさまざまであるので，一つの説明書では，ごく始めの方で，レベルを揃える（即ち同じ線からスタートできるような）予備工作が必要である。もしこれがどうしても長くなるような場合には，説明書を分割して「予備篇」と「本体篇」とする方がよい。

そして「本体篇」には冒頭に読者は既にこれこれのレベルに達していることを要する，と条件を明示しなければならない。いわば大学における「既修課目」の明示と同じである。例えば「予備篇」を読了していること」とか，「某氏著「…入門」理解程度のレベルにあること」などとすればよい。

そこまで厳重にできない場合には，本文の始めに「予備知識の整理」とでもいうような複習のための章節を設けて，手短かに必要な概念や事実を列挙し，簡短な定義や解説を与えることが望ましい。

また一事項を説明するに当って漠然と「よく知られているように」とだけ書くのでは説明にならない。また「既に定義したように」というのも，どの章節にとか，別の説明書のどの部分にとか，明瞭に示さなければならない。

(2)　引用・参照を容易にすること

前節に述べたことを実行するためには，一つの説明書の中のある論述を他の部分で確実に引用する必要があり，このためには章節さらにはその下位の区分に至るまで，明確な区分を判りやすい区分記号で示さなければならない。

通常 1. 2. 3 のような区分記号が使われる（ＪＩＳ　Ｚ参照）が，他の部門に

くらべ，ずっと多段の階層を使わざるを得ない。このた記号全体が重苦しくなることもあるので，更に一工夫必要であろう。

また重要な定義，特性等の記載に，上記の章節区分とは別に，番号を振ること（数学において行われる「１・１２式」のような記載法）も活用が便利であろう。

(3) 個条書き，表，図の一層の活用

よく知られていることであるが，一層の活用が望ましい。場合によっては本文の一般文章による記述と重複しても差し支えない。また重要な事項，式等を，ところどころに一括して示すことも使うとよい。

(4) 飛ばし読みを可能にする措置

説明者は論理正しく抜けなく書く努力をすればするほど，短時間で要点だけをとらえることが困難になる。実用上には，説明書全体を読まず，ある観点から見て必要な事項だけを飛ばして読み取ることが必要になる。

これを可能にするためには，全文を読んで全体を知りたい読者と，特定の重要事項だけを知りたい読者とを区別し，特に後者のためには，この部分とこの部分だけを読んでも理解ができる旨を明示することが望ましい。

あるいは初めから，こういう目的の者はこれとこれだけを読め，という指示をして，始めから複数路線を容認した書き方をとることもできる。ただこの方法は著者にとってはやや負担の増加になる。

思い切った行き方は，同じ製品の説明書を複数作り，「何々を目指す人用」と断って，路線別に説明書を分ける方法である。もちろん作成者の負担は増すが，利用者には勧迎される。これを徹底すれば，マスター版をまず作り，利用目的別に，マスター版中の必要な部分だけを拾い集めて，特別版を何通りでも作る（コンピュータ編集，ワードプロセッサの利用によりこれは可能となる）ことになる。

(5) 頻繁な改訂を予想した処置

まず説明書の始めに今までにいつ改訂が行なわれたかの経過を個条書きにした記述がいる。たとえば以下のような体裁でよい。

第１版　年月日

第２版　年月日　２章，3.5を改訂追加

第３版　年月日　全般にわたる改訂

第４版　年月日　1.8，3.4／3.6を改訂

これは通常励行されている。

次に改訂のあった章節の始めに，この項にはいつ・どんな形の改訂があったと記載する方法である。

(6) ページ付けの問題

改訂によりページ単位の差しかえですむ場合には問題ないが，前の１ページが改訂後２ページ以上に相当することとなり，一連番号以外のページ付け（例えば旧２１ページに対し２１ａ，２１ｂ等）が必要になる。ただしこのような場合には追加ページがどれだけあったのかは，自動的に判らない（上の例では２１ｃページがあるが脱落しているのかもしれない）ため，本文の文脈で判断するほかないが，これはしばしば危険である。

従って新追加ページから旧ページに移る際には，新追加ページの最後に，旧ページのどこに飛ぶかの指示をページ付けの枠の中でしなければならない（例えば２１ｂページに，次２２という指示を入れる）。慣行すれば簡単なことであるが従来は行われなかった。

C 4. ドキュメント検索の便宜

ソフトウェア分野ほど説明書が膨大になるところはない，というのが定説になっているが，その膨大さを克服する手段が講じられていないこともまた一つの特質ではあるまいか。

これはドキュメント即ち資料が広く流通する公刊物になっておらず，発行元が図書資料出版（あるいは販売）の専門業でもないので，この面の配慮に欠けているものと思われる。

従って，日本中に，どれだけのソフトウェア関係のドキュメントがあるのか，目録もなければ，入手経路も整備されていない。従って急務は業界全体の努力によって，各メーカの配布可能なドキュメントの目録を作り，これを絶えず更新してゆくことであろう。もちろん，近い将来この目録がデータベースになり，オンラインで自由に検索できるようになることも望ましい。

この一般的な目録の他に，ユーザ機関が自己保有のドキュメントの現状を常に新しく確認する作業が必要であるが，これをなるべく僅かなマンアワーで実行可能にするため，最もよい（発行元の行える）サービスは，各ドキュメントにやや厚手の１ページを設け，これに一定形式で，直ちに目録に使えるようなカードを印刷して提供することである。このようなサービスは，一般の技術資料ではある程度行われている。

このページからカードを切取り，カード箱に入れ，利用者の希望する順序で配列すれば（例えばメーカ別，ドキュメント番号別）直ちに自分の手許にカード目録ができる。カードの裏面には，旧版との関係，内容ページの差し換えの日時等を記入する欄等を設けておき，利用者の僅かの手間によってアップデートが可能なようにしておけば，目録の維持も容易となる。

ドキュメントの存在を検索によって知り得たとしても，ドキュメントの現物を必要の都度入手することに時間がかかってはまた意味がない。即ち一次資料の保管を，ドキュメントそのものの形で行うか，または他の形態（例えばマイクロ形式）で行うかも考慮の対象となろう。とくにドキュメントが大きく数も多くなれば，物としての取扱いにも困難が生じ，印刷形式は差し換えにも難点があるので，発行元がマイクロ形式での供給することも十分考えられる。航空機整備における図面や仕様がすべてマイクロ化され，メーカから整備センターに迅速に送られ，保管も容易である現状には注目すべきであ

ろう。

D　TC—46側の援助

以上のいくつかの示唆または提案に関してＩＳＯ／ＴＣ４６は既に多数の規格を提供できる。またＴＣ４６の受持範囲で各国の規格が存在し，いまだＩＳＯ規格となっていないものもある（例えばTechnicalmanalの作成方針，イギリス規格ＢＳ４８８４，Part１－１９７３参照）

E．導入面

ソフトを導入する際まず考えなければならないことは，そのソフトがどこまでの機能をもっているかを見きわめなければ導入出来ない。又導入によってどれだけの経済的な利益が得られるかその時期はいつかも併せて見きわめる必要がある。

導入のステップを企画立案，実行計画，導入の３つのステップに分けその各々で何を標準化し実施確認すれば期待した効果が十分あげられるか述べる。

Ｅ１．企画立案

まず目的ー期待する効果ーを明確にすること，そのために現状業務の実態を正確に把握することが基本となる。すなわち対象業務ー人事管理，生産管理，経営管理等どの業務システムをどの様に改善するのか，扱う情報形態は何か一文字，図形，音声等，関連部門はどことどこか等情報の入力から出力，その利用に至るフローを作り実態を整理する。

次に経済比較を行う。ハードの費用，消耗品，保守費用，スペース費用，人件費，ソフトの費用，運用経費等を仕事の量にリンクさせ算出し，利益が出るか否か検討する。その際情報処理のスピードアップ，精度アップ等も利益計算に含められる様考慮することが大切である。

Ｅ２．実行計画

導入により期待する効果が得られることがわかったら実行のための計画をかためることが必要である。

(i) 金物，システムの選定

(ii) 資金計画

(iii) 人員計画

(iv) データ体系（コードの整備，マスターの作成）

E 3. 導入のステップ

実行計画にそって担当部門が次の各項目につき１つ１つ関係部門と検討，具体化する。

(i) 本体，端末，インタフェースの詳細，仕様の確認

(ii) 設置条件

- 電源
- スペース
- 回線
- 環境
- 重量
- 機密保持
- 防災設備

(iii) 搬入計画（搬入口、移効手段）

(iv) 設置

(v) 試運転

基本ソフト，アプリケーションソフト，その他必要なソフトで試験を行う。

- 機能確認
- 異常処理
- 精度
- 容量
- 処理能力（実行時間，レスポンスタイム等）
- 信頼性（致命故障，不良率，稼動率）
- 安全性（エラー，リカバリー）

○　操作性（マニュアルの質，運転性，マン・マシンインタフェース，室温，騒音）

(vi)　検収

保守資料，取扱説明書，操作手順書，その他必要な資料の確認と受理。

(vii)　保守契約

契約書，保守部品，メーカ無償提供部品の確認，サービデポの確認，故障時のペナルティ，復帰迄のルール化，バックアップ体制，消耗品の供給体制

(viii)　教育

○機械操作

＊電源投入，切断手順の基準

＊オペレーション基準

＊異状時の操作手順

○安全対策（機密保持）

＊防災等に関する基準

＊防犯に関する基準

○アプリケーションソフト作成

以上ハード，ソフトの仕様確認から教育迄の一連の作業を確実に実行し導入が完了となる。

E 4.　マン・マシン・システムとしての諸問題

効率よく利用するためには，マン・マシン・システムにおいて次の点を事前に十分考慮する必要がある。

(i)　システムとして必要な冗長性を持っているか

(ii)　フール・プルーフ，フール・セイフが考慮されているか。人間の誤った操作に対して安全サイドで事態の進行をくい止めるシステムが出来ているか。理想的には誤った操作をしようとしても，その方向には動かせないようなシステムか否か資料，環境仕様で明確にしておくこと。

(iii)　フェイル・ソフトリーが考慮されているか，局部的な故障に対してシ

ステムに直撃的な影響を与えないように局部的に機能を失ってゆくシステムか否か資料，環境仕様で明確にしておくこと。

(iv) 診断がやり易くなっているか資料，環境仕様で明確にしておくこと。

E 5. 品質保証と標準化

導入時，要求する品質が十分に発揮されるシステムか否かを確認することはきわめて重要なことで，そのために少くとも次の事項に対しては，用語，表現方法，単位，評価基準を標準化しておく必要がある。

- ○機能面
  - 規　格
  - システムの形態
  - 異常処理
  - 汎用性
- ○性能面
  - 精　度
  - 容　量
  - 処理能力（ソフト，ハード）
  - 限界性能
- ○信頼性
  - 処理能力の低下
  - 故障率
- ○安全性
  - エラー，リカバリー
  - システム再構成
  - 財産の保護
  - P　L
- ○保全性
  - 保守体制
  - 保守マニュアル

○ 操作性
- ユーザミス対策マニュアル
- （運転準備から終了までの操作手順）
- マンマシンインタフェース
- 環境

○ 機能追加の場合の注意事項と手順

○ 機能修正の場合の注意事項と手順

## F　利用者側の管理・運用面

ソフトウェア製品を導入したユーザが，それを効率的に利用して行くための運用・管理面から要請される標準化項目について考察する。ここで云う運用に関与する人としては，情報処理部門の専門要員（システム・エンジニア，プログラマ，オペレータなど），管理者のほか，一般業務部門で端末からコンピュータを直接利用するエンドユーザやパーソナル・コンピュータのユーザなども含まれる。

### F 1.　マニュアル

マニュアルの不親切，不備を指摘する声は多い。望まれていることは，内容が正確・完全で，必要箇所を探しやすく，かつ記述が理解しやすいことである。これに関し，改善すべき点は次の通り。

(i)　分類体系

説明内容によって，例えば（H社の例）

概説書

解説書

文法書

操作書

手引書

その他

のような分類が用いられている。各社の現状は大同小異なので，これらの呼称と各々の記載事項を統一する（例えば，概説書には所要リソースの概算法を必ず含めるようにとか）ことは可能であろう。勿論，ソフトウェアによっては，これらが独立の冊子とならず，合冊ということもあろうが，それは差支えない。

この分類によって，それぞれが対象とする読者の種類は規定されるが，更に読者層を技術レベルによって区分（例えば，初級，中級，上級）して，

各レベルに応じたマニュアルが，そのような区分を明記して用意されることが望ましい。（レベル別のマニュアルを用意しない場合には，少くとも初級者に対する読み方のガイドをつけて，どこどこ読めばそのソフトウェアが使えるかすぐ分るようにしておくべきである。）

ついでに，この分類体系に狭義のマニュアルだけでなく，

教科書・教材

カタログ

参考資料

を加えておくと，整理上都合がよいように思われる。（教科書と初級マニュアルは区別がつきにくいところがあるかもしれない。）

(ii) 編集方法

前項で各マニュアルに盛るべき内容が定まったとして，その項目だてと記載順序を同種のものについて統一することが望ましい。あらゆる種類のマニュアルに対して規定することは難しいが，言語やアプリケーション・パッケージの文法書における各命令（又はコマンド）の説明は，例えば，

命令（コマンド）の名称

機能の説明

書き方（又は指示方法）（形式）

オペランド（又はパラメータ）の意味と規則

使用例

異常のチェックとその表示及び処置法

その他注意事項

のように整理して記されていることが多いから，ひな型（基準案）を設定することは可能と考えられる。

それから，検索・参照の便について特に配慮が必要である。

(iii) 用語

ＪＩＳ情報処理用語の使用は当然だが，アプリケーション・プログラム

の機能に関しては，当該アプリケーション分野のＪＩＳ用語（例えば，オペレーションズ・リサーチ用語とか，品質管理用語など）を使用するものとする。そうは言っても，現在ＪＩＳで制定されている語いは少なく，文部省学術用語を加えてもまだまだ不足する。特に，事務関係のアプリケーションにおいて，業務機能や情報処理機能の表現の統一（用語とその定義の確立）を図ることは，単にマニュアルの理解を助けるのみならず，ソフトウェアの部品化とその流通・活用に極めて有効と思われる。

(iv) 文章

読みやすく，理解しやすく，簡潔で（初心者向のものはやや冗長になっても親切で），明快な記述になっていなければならない。あいまいな，あるいは誤解のおそれのある記述があってはならない。これらの点については2.2.3(3)C参照。

(v) 提供方法

マニュアルの改廃が客先において確実に行われる体制を，メーカ側で考えておく必要がある。形態についても従来の印刷製本のほか，マイクロフィッシュとか，テープ等の各種磁気媒体などいろいろ考えられ，またオンラインということも将来の話としてある。ソフトウェアやマニュアルの種類に応じて，適切な方法が選ばれればよい。

## F 2. 教育

先にマニュアルについて述べたことが，教育についてもほぼ当てはまる。まず，教育コースを体系化して，各コースは，どういう人たちを対象に何を教えるのか，つまりそのコースを受講すると何ができるようになるのか，受講にはどのような予備知識が必要とされるのか，他の関連コースとの関係はどうなのか，などの事柄を明らかにしておく必要がある。教育方法や教育内容に立ち入った標準化は，一般的には（基礎教育や，ＦＯＲＴＲＡＮなど標準言語の教育を除いては，）困難と考えられ，当面やれることとしては，教育コース案内書（ソフトウェアのカタログに相当）の記載項目の標準化ぐらいかと思われる。

なお，教育手段に関しては，開催頻度の高いコースにビデオによるＰＩ方式の採用が行われているようだが，将来ＣＡＩ（Computer Aided Instruction）の新たな発展と普及の可能性が考えられる。その場合には，コースウェアの流通促進のため，教育用ソフトウェアのインタフェースの標準化などが要請されるかもしれない。

また，教科書の記述面に関しては，マニュアルで述べたのと同様のことが要請される。

## F 3. 言語仕様，ユーザ・インタフェース

標準化された言語であれ，そうでないアプリケーション・パッケージであれ，いかなるソフトウェアにおいても，そのコマンド，ステートメントや入力データ様式は，生産の立場よりも利用の立場を重視したものとして定められるべきである。エンドユーザがコンピュータを直接利用することが多くなって行くのであるから，仕様決定の際，この面の検討がもっと必要かと思われる。例えば，現行のＯＳのＪＣＬは，エンドユーザ向にはもっと簡素なものが用意されるべきではなかろうか。勿論，高度の利用を考える人には低レベルのインタフェースが必要である。従って，利用者の幅の広がりに応じ，

それなりに完結した複数のレベル（通常２又は３）のインタフェースを設定すべきものと考える。このことは，基本ソフトウェアに限る話ではなく，アプリケーション・パッケージにおいても全く同様である。

このような点に留意した上で，更に次のような標準化が，ソフトウェアの流通促進に寄与するものと考える。

(i) 基本ソフトウェアのユーザ・インタフェースの標準化

ＯＳ，ＤＢ／ＤＣ等の基本ソフトウェアに関して，（将来拡張の可能性を残した形で）ユーザ・インタフェースを標準化（統一化，固定化）する。

(ii) 機能部品としての標準化

汎用の完成品であるアプリケーション・パッケージでカバーしきれない業務処理に対し，今後半成品の利用が進むものと思われる。そのような需要に応じられるように，パッケージ自体を徹底してモジュール化し，各モジュールが機能部品として利用できるように作っておく必要があるが，一方先にａ．(iii)で述べたように，これら機能の定義を標準化し，インタフェースを標準化することが，部品の流通に極めて有効と思われる。科学技術計算の分野では，ＳＳＰ（ＩＢＭ）以来若干のサブルーチン・ライブラリが流通しているが，その名称，引数の書き方の統一が望ましい。事務計算の分野では，今後前記の標準化を進める必要が大きい。

(iii) 現在かなり普及しているアプリケーション・パッケージのユーザ・インタフェースの標準化

例えば，離散型シミュレーション言語ＧＰＳＳ（ＩＢＭ）は，多くのメーカから提供されるようになっているので，言語仕様を標準化した方がよい。このようなものは，ＤＹＮＡＭＯ等他にもいくつかある。

また，主として科学技術計算の分野のことになるが，入力データ様式だけを統一し，あとは自由に競争させるのがよいものもあろう。例えば，ＬＰコード（線形計画法）では，かつてＳＨＡＲＥが統一様式（ＩＢＭユーザ内だけでの統一だが）を定めたし，このようなことは，他にも統計解析

構造解析などいくつかのアプリケーションで考えられる。

(iv) プログラム言語の標準化と改訂

まだ標準化されていない言語を標準化することと，既に標準化されている言語の日本語処理を中心とする拡張である。

以上何れも日本独自の標準化を行うことは適当でなく，国際的な標準化活動と一体化した形で，その動向をにらみながら，国内の標準化を進めて行かねばならないので，なかなか結論の出しにくいところがあろう。現在開拓の遅れている(ii)の事務分野での実施あたりが，そういう面ではむしろ手をつけやすいところかもしれない。

## F 4. 操作性

マン・マシン・インタフェースに人間工学的な配慮がもっと必要だということで，いくつかの指針がまとめられようが，将来標準化と結びつきそうなものとしては，メッセージの種類（緊急度）とカラーや警告音の使い方などが考えられる。

また，対話型のコンピュータ使用の普及に鑑み，標準プログラム言語等において，ステートメント，コマンド等の省略形（入力用）も標準化しておくことが望ましい。

## F 5. セキュリテイ（プライバシーを含む）

故意又は過失により，正当な権限を有しない者がコンピュータ・システムにアクセスして，システムを無断使用あるいは破壊し，また記憶された情報（データ・プログラム等）を消失，改ざん，漏洩などすることから，システムを保護することは，今日極めて重要な問題となっている。このような事故の発生は，官庁，公共企業（証券金融，交通運輸等を含めて），情報処理サービス業等は言うに及ばず，一般企業においても，経済・社会活動に大きな混乱をまき起すおそれがあり，また不正入手データによるプライバシーの侵

害や二次犯罪の可能性をはらんでいるからである。

これに対し，ソフトウェア面からの対策として，次のようなものが実施又は提案されている。

① ユーザＩＤ，端末番号等とデータ又はプログラムに関する権限リストによるチェック（バッジカードの使用を含む）

② パスワード（合言葉）や暗証番号によるチェック

③ 情報の暗号化

④ 実績記録による事後チェック

このうち，①，②はふつうに行われているが，③については，我が国でも，米国等の状況をにらみながら，標準方式制定の検討を進める必要がある。④の記録のとり方には，いくつかのレベルが考えられる。

そこで，まず行うべきことは，セキュリティのレベルを設定し，各レベルに対応する対策手段を整理する（後述の障害対策と同様に，ＯＳ等との役割分担の設定を含め）ことである。このような標準が定められていると，データ（又はプログラム）の重要度と保護のためのコストを勘案して，それぞれのデータ（又はプログラム）に適当なセキュリティ・レベルを選ぶことができる。

なお，システム監査は，この問題に対しても行われるべきであるが，そのための手引やツールを整備することが望ましい。

## F 6. 障害対策

障害発生に備えて，チェックポイントを設けて，その時の状態を保存したり，その後のジャーナルを取得するなどしておき，一旦障害が発生したときには，それらのデータを用いて状態の回復をはかり，処理の再開が可能となるようにしている。このような障害対策に関していくつかのレベルを設定し，各レベルに対応する対策手段を標準として整理しておくことが望まれる。同時に，障害対策（回復手順を含む）におけるＯＳ，データベース管理システ

ム，通信制御プログラム，アプリケーション・プログラムの四者間の役割分担の基準を設定することも必要とされている。

## F 7. 評価

ソフトウェアの評価は，機能，性能，使いやすさ，信頼性，経済性，将来性等の各面からの評価をライフサイクルにわたって総合して行われなければならない。そのような評価作業は，発注時に資料，テスト，調査と予測をまじえて行われるが，ここでは更に導入後，実績データに基いてなされるフォローアップに関して述べる。この結果は，当該ソフトウェアの改善，マニュアル，教育法等の改善，ユーザの利用法やオペレーション面の改善のためフィードバックされる。

(i) 機能

もし先に述べた機能定義が標準化されていれば，論理的な機能内容としては，ほぼ事前に評価できている。運用してみて初めて分るのは，性能と組合せての実用的な機能範囲とか，複合的な機能，補助的な機能，異常処理などの点であろう。

(ii) 性能

ソフトウェアの性能を表すのに，スピード，リソース使用量等に関し多数の指標がとられているが，それらの数値は使用環境の諸条件によって大きく変るため，性能の表示は難しいとされてきた。しかし，主要な性能指標の厳密な定義とその測定法の標準化は是非とも必要である。

なお，科学技術計算においては，結果の計算精度も重要である。

(iii) 使いやすさ

この言葉を比較的狭くとらえても，コマンドやステートメントの使い方（理解のしやすさと機能の良さ），入力のやり方（様式記入又は入力操作のやりやすさ），出力の整理の仕方（見やすさと活用のしやすさ）等が含まれるほか，少々のことで倒れない，間違いを親切に指摘してくれる，使

用リソースの制約が少ない，処理時間が手頃でコストが安い，などがある。更には，初心者にはバカチョン的だが，なれたユーザには高度の機能を提供できるとか，一部の機能だけ取出したり，他と組合せたり柔軟に利用できるとかも望まれる。これらの項目のうちには，既に他の節で触れたもの（性能等）もあるが，使いやすさはそれらを含めての総合評定で，利用者のレイティングによることとなろう。

(iv) 信頼性

ソフトウェアについても，信頼性の管理指標を標準化して普及させるべきである。なお，テスト基準の設定が望まれる。

## G 保守面での標準化（含管理面）

ソフトウェアを効率よく保守するための標準化すべき項目を抽出するにあたって，まずソフトウェアの保守作業の内容を概観し，その特徴を明らかにしてみる必要がある。

### G 1. 保守作業の内容

ソフトウェアの保守作業は運用中のソフトウェアに関して，プログラム，ドキュメント，データなどを運用条件の変更に応じて修正，変更するものであり，これらは維持・管理作業とも呼ばれている。また，ソフトウェアが設計，製造され，運用に供されると同時に，保守に引き継がれる場合のソフトウェアの品質，並びに保守期間中に発生する事象をとりあげてみるとつぎのようになる。

まず，ソフトウェアの品質について見れば一とおりのデバッグ作業，性能評価などが行われているものの，内在する潜在バグや変則的な運用で発生する一時的な処理能力低下などがあげられる。また運用マニュアルなどのドキュメント類の誤記，記述の不備等が明らかになってくる。つぎに，保守期間中に発生する事象としては，運用をより円滑に行うための，各種コマンドの

追加，修正，入力手順や出力形式の変更に伴うプログラム修正，業務内容の変更に伴うプログラムの部分修正などがある。さらに，ＯＳやハードウェアの陳腐化に伴うバージョンアップに伴う機能，性能確認作業などがある。

このような保守作業をＥ．Ｂ．Swanson （１９７６年）は修正保守，適応保守，完全化保守として細分化し，表Ｇ－１のとおり分類している。

特に言えることは，ソフトウェアの保守作業の内容は，現時点では必ずしも明確になっていないことであろう。

表Ｇ－１ソフトウェアの保守作業

| | 修正保守 | 適応保守 | 完全化保守 |
|---|---|---|---|
| 作業内容 | a.プログラムの処理エラーの修正（潜在バグ）<br>b.処理能力エラーの修正<br>c.ドキュメント，マニュアルの誤記，不備の修正 | a.データの環境変化に対する変更（祝祭日の変更，利率，税率等の変更<br>b.業務内容の変更・追加に伴うプログラムの変更・追加<br>c.処理環境変化に対する変更（ハードウェア，ＯＳ等のバージョンアップ） | a.処理効率の向上やメモリ節減<br>b.機能洗練（入力手順，出力形式，出力情報追加）<br>c.運用効率向上（ファイル統合）<br>d.利便性向上（コマンド形式，追加）<br>e.保守性向上（診断プログラム変更，ドキュメント変更） |

## Ｇ２．保守作業と特徴

表Ｇ－１に掲げたそれぞれの作業を掘り下げてさらに詳細に眺めてみると，つぎのような特色がある。

修正保守の中でのプログラム処理エラーや，処理能力エラーの修正は，そ

の発見や，修正にきわめて高度な技術を要する。これは，すでにデバッグ工程を終え，その段階で考えうるエラーはすでに修正が終わっており，残っているエラーは稀にしか使用されない機能モジュール上のものや想定されなかった運用条件で発生するもの，さらには複雑に組み合わさった条件でしか起こらないもの，ハードウェアの局部的，一時的な故障が結果としてプログラムエラーとして検知される場合などがある。そしてこれらの原因を解析していくためには，ＯＳ，アプリケーションプログラム，運用条件，ハードウェア等の機能動作に熟知し，これらの個々の誤動作が原因となって生じる結果を予測してエラーのある場所を特定していかなければならないからである。

さらにこの切り分け作業を短時間に行いエラーを修正するプログラムを設計，製造し，組み込み後の機能や性能面に与える影響の確認なども短時間で行う必要がある。場合によってはこのエラーをパッチプログラムで対処することもあり，パッチ管理表などのドキュメントの整備も必要となる。

適応保守では業務内容の部分的な変更・追加に伴うプログラム開発，ハードウェア，ＯＳの変更に伴う，プログラム整合性の確認が比重の大きい作業である。この場合のプログラムは一般的には規模の小さいものであるが，保守の状態に移っている環境下でのプログラムの開発は，その他の作業と並行しながら，従来のプログラム設計思想を踏襲し，現行プログラムとのインタフェース条件と整合をとり，かつ性能を維持するという制限を克服して開発作業を進めなければならない。とくにこれらの作業はその発生を予測できないところが宿命的な問題である。ＯＳやハードウェアは，年を経るたびに新たな機能付与や，技術的陳腐化による更新が行われ，特にＯＳ類は，長期的な保守効率を低下させないためには，更改に追従していかざるを得ない背景もあって，乗せ替えが行われている。ＯＳやハードウェアの更改に伴なって，現行ソフトウェアシステムとの整合性確認が不可欠となってくる。この作業は，内容，量ともシステム開発時と同等のものとなる。

完全化保守作業は，実運用とシステムのチューニング作業とも言えるもの

で，利用者からの種々の要望に応えるためのプログラムの改造が主体となる。これらの作業は，システムの運用開始初期に集中し，その後は除々に減少していくものである。そのほかに運用との対応で保守を担当する部隊はその窓口となって雑多な作業の対応を余儀なくさせられる。

このようにソフトウェアの保守作業はシステムを円滑に運用していくために重要な役割りを持っているわけであるが，その大きさを，情報処理システムのライフサイクル中の費用比で考えてみると，最近の大規模システムの例では全体の６０～７０％を占めるようになっているという報告があり，それだけに，その効率化が強く望まれている。

## G 3. 保守面から標準化すべき項目

ソフトウェアの保守作業の内容と特徴で述べたように，ソフトウェアの保守作業はプログラムの開発，機能検証，性能評価などであり，これらは開発段階の作業と同様となる。したがって保守面におけるソフトウェア標準化項目は開発段階のそれと重複するものがほとんどであるが，主として保守面での問題から標準化する必要のあると考えられるものを表G－２にあげてみた。

表 G－2 保守面における標準化項目

| 分　類 | 問　題　点 | 標準化項目 |
|---|---|---|
| 全　　般 | 保守面で行うべき作業内容が不明確であり，作業量，費用の見積りが困難となって，要員確保，予算化，保守費用回収などの問題となっている。 | 保守作業内容（作業標準） |
| | 保守作業内容があいまいなので，追加プログラムの作成費用，改造修正作業，運用支援作業などの費用負担区分などの契約もあいまいとなっている。 | 保守契約内容 |
| ドキュメントの保守方法 | 修正保守，適応保守，完全化保守作業で必要ととなる現行ソフトウェアの機能，構成，性能インタフェースに関するドキュメント，運用マニュアル等の管理主体と管理方法や保全作業から見た記載方法，変更方法などがあいまい。 | 管理すべきドキュメント<br>ドキュメントの管理方法<br>各種ドキュメントの記載内容 |
| 修正，追加プログラムの開発ツール | 保守作業全般で修正，追加を行うために開発段階で使用したツール類と関連ドキュメントが保守に引きつがれない。 | 開発用ツールとそのドキュメントの保守方法 |
| 性 能 確 認 | 修正，改造等による性能確認方法が確立されていない。 | 性能確認のためのガイドライン |

しかしながら，ソフトウェアの保守面における種々の問題は最近になってクローズアップされてきたものであり，今後，これらの問題が工学的見地から各種の試みが行われようとしているのでその成果を待って標準化に着手した方がよいと考えられるものも多い。

## H　拡張面での標準化項目

運用中のソフトウェアは，業務内容の変更や新規業務の追加などの周囲情勢によって機能の追加変更は頻繁に発生する。このような追加変更に容易に

対応できるようにするために，拡張性を十分考慮したソフトウェアの設計が重要である。

拡張性を確保するということは，一般には作業テーブルを大きく取る，コマンドの指定領域を大きくする，機能をモジュール化し，他のモジュールとのつながりを少なくするなど，かなりの冗長設計を行うこととなり，従来はメモリなど高価なハードウェア資源の制約等からあまり積極的には行われなかったとも言える。最近では集積回路等の飛躍的な進歩でハードウェア価格をあまり意識しなくて済み，高騰するソフトウェアの費用を抑えるために，容易に追加変更できるようなソフトウェア設計が重視されている。

拡張性確保のために必要な標準化項目はソフトウェアの設計工程で行われるすべての内容に渡るような基本的なものばかりであり，これらを具体的に挙げると表H－1のようになる。

表H－1 拡張面で標準化が必要な項目

| 分　類 | 問　題　点 | 標準化項目 |
|---|---|---|
| コード類 | 識別項目の追加に対して所要桁がふやせない | コード設計と拡張性 |
| テーブル | 業務プログラムの追加，入出力機器の追加をする場合，該当テーブルの余裕がない | テーブル設計と拡張性 |
| ファイル | ・データ項目や属性追加のためのエリヤがない<br>・データ構造が統一されていない | ファイルのデータ構造と拡張性 |
| コマンド | 新たなコマンド追加が必要な場合，OS等の制限から容易に追加できない | 拡張を考慮したコマンド設計指針 |
| プログラム構造 | 機能の追加修正に伴うプログラム変更に対して，関連するプログラムが入り組んでいて容易に追加修正ができない | モジュール間の独立性を保つプログラム構造設計 |
| ドキュメント | 機能拡張のためのインタフェースとなる項目の記述がない | ドキュメントの記載項目の標準化（拡張用インタフェースの項目明示） |

## I 契約面

ソフトウェアの流通を円滑に進めるためには，これを一つの製品のように，その機能，品質などの仕様と価格，納期，アフターサービス条件などが明示され，これに基づいて購入や賃貸ができるようにする必要がある。

ソフトウェアには多くの利用者が共通的に利用できる汎用のものと，個別の利用者の業務用として専用に開発するものがある。前者は一般の製品のような取り扱いができるが，ソフトウェアが容易に複写できるので売買契約等の中でその権利を保全する方法がとられる。後者の場合には予算化の問題などから開発に先立って総費用の見積りを行い，価格，納期，品質などを取り決めて契約するという特注品のような扱いとなる。また，ソフトウェアはシステムに組み込んで始めて機能するものであり，その組み込み作業の技術的支援作業やその後のバージョンアップソフトウェアの提供，保守条件などが明示される必要がある。

以下に契約面における標準化項目を検討するにあたって現状とその問題について概観してみる。

### I 1. 価格積算

契約における最も基本的な項目は契約に係る費用であるが，ソフトウェアの場合はその価格の積算方法が確立していないことである。

ソフトウェアが製品として買売されるためには価格が客観的な形で定まる必要がある。対価の算定はソフトウェアの設計，製造，及びそのドキュメント作成などに要した費用とデバッグ等に使用した計算機の減価償却費，利益等を計上する方法が一般にはとられている。しかしながら，これらの費用は，特注品と既製品とでは扱い方が異なってくる。特注品のソフトウェアの場合は通常，システムの大まかな条件が決まった段階で，しかも受注競争の果てに契約が行われることとなる。さらに，システムの詳細な検討が進むにつれて，当初に想定されなかったような条件が明確になってくる場合がしばしば

であるがこれらのために開発する追加プログラムも契約金額の範囲で実現することになってしまい，結果として出来上がったソフトウェアの規模と費用との関係が大幅に狂っていくこととなる。これに比較して，既製のソフトウェアは多少様子を異にしている。既製のソフトウェアは多くの利用者を対象とした汎用のもので販売する時点では既に開発が終っているので開発に係るすべての所要経費，商品性，需要数，利益等を見込んで価格を設定することができる。しかしこの価格には，商品性，すなわち，そのソフトウェアの付加価値と市場ニーズという要素が含まれているために，一律に価格設定の法則を見出すことはできなくなっている。

このように，ソフトウェアを製品として扱うには種々の問題を含んでおり，さらに契約行為が当事者間の合意のうえで行われるという原則に依っているだけに問題を一層難かしくしていると言えよう。しかしながら，ソフトウェアの販売を円滑に進め流通の促進を図るためには，標準的な価格設定方法が不可欠である。

### I 2. 納期

納期が関係するのは特注品の場合だけである。価格積算と同様に，納期についてもソフトウェアの開発にかかる作業量を見積って設定されるが作業の見積り誤りや作業遂行上の手戻りなどがあって約束された時期に納入が行われなくなる場合がある。

納期の遅れに関して一般の契約では遅れ分に対するペナルティ制度が適用されることが多いが，特注のソフトウェアの場合，受注者側の責に帰するか，発注者側の責に帰するかの判別がつかない場合があり，ペナルティが適用されるケースは少ないのが現状であろう。

納期の取り決めは作業を予定通り進めるための原動力となり，所要時間に，所要の物が納まり次の作業に着手できることとなる。もし紛糾しても，解決の糸口を容易に見出せるためにも，納期の設定は，契約上重要な項目である。

## I 3. 品質

ソフトウェア製品において最も重要な項目が品質であろう。ソフトウェアの品質の指標は処理エラー，すなわちバグの数である。内在するバグはソフトウェア納入後の後続作業の進捗の大きな障害となり，この品質が実質上の納期を決めてしまうといっても過言でない位大きな意味を持っている。

製品として捉えた場合のソフトウェアは適用するシステムに組み込んで始めて機能するものであり，この意味においては半製品と言うことができる。半製品の機能検証方法はシミュレータなどのツールを使って行われることが多く，このシミュレータが如何に実システムの機能条件を模疑し得るかが内在バグの減少につながっている。この機能条件の模疑のやり方については，検証作業の従事者の資質に依存することが多く，経験豊富で洞察力をもった従事者の作業によってのみバグを多く検出せしめ，品質の良いソフトウェアが完成する。言い換えれば，条件の設定されなかったプログラムのバグは修正されず，内在したまま納入されてしまうということである。したがって，契約上この品質表示はもちろんのこと，品質確認方法に言及することが重要であるがこの品質確認方法の客観性が維持できるよう，その標準化が早期に行われることが必要である。

## I 4. かし条項

一般の製品の売買契約では製造者の無過失かしを許容して，その保証期間を商法では製品引き渡し後6ヶ月，民法ではかしを発見後1年と定め無償の修理を義務づけている。ソフトウェアにおいてもこれを製品として扱う場合には同様に契約にかし条項を設ける必要があるが，残存バグは数年間経過して発見されることもあり，どのような場合にかし条項が適用されるのかについては定まった解釈がない。このため，法制的見地からの十分な検討を要すると考えられるが早期に適用基準が必要な項目であろう。

## Ⅰ5. 保守条件

一般の製品の場合の保守条件はサービス窓口の案内，故障修理の有償，無償の条件などが挙げられている。ソフトウェア製品の場合は，その後の機能追加，性能向上などのバージョンアップが行われるのが特徴であり，購入者の立場からはこれがどのようにして取得できるのかが大きな関心事である。

また大規模なソフトウェアなどでは別途保守契約を結ぶ場合もあり，購入契約における保守条件と，保守契約で行われる条件との正別が明確になっている必要がある。

## Ⅰ6. 再販売権

今まで述べてきたソフトウェアは単一製品として扱った場合における問題であったが，汎用のソフトウェアではそのプログラムを全体の一部として組み込み，さらに付加価値を付与してシステム全体で販売されるケースが多い。これはソフトウェアの単品の売買による流通に加えて，再販売によるソフトウェア流通という二大流通路形式のために大きな意義がある。

再販売を可能とするためにはソフトウェアの再販売権の取得契約を結ぶことになるがその権利取得対価，再販売実施条件などについて，単品の売買契約に順じて標準的契約条項を定め，再販売によるソフト流通の促進を図る必要がある。

以上のとおり，ソフトウェアの購入契約に関連してその主たる点について標準化が必要な項目を挙げてみたがこれを表Ⅰ－1に再掲する。

表Ⅰ－1 契約面で標準化すべき項目

| 分　類 | 問　題　点 | 標準化項目 |
|---|---|---|
| 価　格 | ソフトウェアの規模・内容と価格の関係を客観的に裏付ける指標がなく見積価格の精度が悪い | 価格見積りのためのガイドライン |
| 納　期 | ソフト開発に必要な工数の適切な見積り，納期設定の精度が悪い | ソフトウェアの規模内容に対応した必要工数の見積りのガイドライン<br>工数と納期設定のためのガイドライン |
| 品　質 | 機能検証従事者の資質によって，表示品質の内容に差が出る | 機能検証方法のガイドライン設定 |
| か　し | 残存バグの検出時間とかし期間の関係を明確にし，かし条件適用の指針を定める | ソフトウェア製品のかし期間の設定法<br>かし条項の適用基準 |
| 保守条件 | 保守条件として一般製品の内容に加えて，ソフトウェアのバージョンアップなどを考慮した保守条項を定める必要がある | 保守条件の明示のガイドライン |
| 再販売権 | 権利売買価格の客観的指標がない。また，品質，かし，保守条件について単品購入に順じた検討が必要 | 権利売買価格設定法<br>再販売権売買における品質，かし，保守条件の標準化 |

# Ⅲ　ソフトウェアに関連して標準化が望まれるその他の項目

## Ⅲ　ソフトウエアに関連して標準化が望まれるその他の項目

Ⅱにおいては，ソフトウェアをプログラム及びそれに関連する文書とし，すなわち，ＴＣ９７の範囲に限定するものとし2.2.2の視座から，2.2.3において具体的標準化項目を述べた。

当委員会では，これらの項目以外についても多くの話題が提案されたので，それらをここに簡単にまとめて示す。

(1)　商法上の問題

- ソフトウェアの著作権
- レンタルの場合の所有権

(2)　経理・会計法上の問題

- ソフトウェアの原価計算法
- ソフトウェアの原価消却法
- ソフトウェアを改訂した場合の付加価値の価格計算法

(3)　ＴＣ４６の問題

- 各種帳票の標準化

(4)　コードの問題

- 商品としてのソフトウェアのコード

### A　コスト

ソフトウェアの標準化項目の中で，必要ではあるが非常に難しい項目のひとつにコストがある。結果として，今回の標準化項目の検討の中では，標準化項目としてはっきりした項目が設定できなかったが，今後の検討の方向について述べてみたい。

コストは，ソフトウェアの設計時に発生する設計コスト，実際にソフトウェアを作成するための製造コスト，そして，保守のために必要となる保守コストに分けられる。

#### A1　設計コスト

設計コストは，開発規模，開発期間によって左右されるが，さらに，特に注文品の場合には，開発の比較的初期の仕様の不確定な段階での見積りが要求されるため，仕様の確定度合，仕様の追加・変更の状況によっても大きく変動する。

(i) 開発規模

普通，開発規模が大きくなるに従って単位当りのコストは高くなる。ソフトウェアが複雑になり，従って，テスト時間，チェック工数等が規模が大きくなるにつれて増加するからである。開発規模は，一般に工数あるいはステップ数（ステートメント数）であらわされるが，一例をあるソフトウェア会社の価格表でみると，規模による1ステップ当りの単価は次のようになっている。言語はコボル，1～400ステップの1ステップ当りの単価を1とした場合の単価の比は

| | |
|---|---|
| 1　～　400 | ……1 |
| 401～　800 | ……1.25 |
| 801～1200 | ……1.5 |
| 1201～ | ……1.75 |

となっている。

開発規模のコストへの影響は，基準となる開発規模を工数またはステップ数で定め，規模による係数を実績データの解析によって定めるような大まかな方向で検討したら良いのではないか。

(ii) 開発期間

設計コストを見積る場合，標準的な開発期間にくらべて開発期間が短い場合はコスト高になることに注意せねばならない。普通，開発期間が短いと仕様の詰めが甘くなり，後述する変更等が多くなる。また，しわ寄せが開発の後半にくるためテスト，トライアル等に無理がきて思わぬロスが発生することになる。規模に見合った開発期間がとれない場合は，例外と考えて標準化から除いても良いが，開発実態で考えると何らかの算式を設定

したいと思う。特に，注文品の契約の際には必要であろう。

(iii) 仕様の確定度合

設計コストは，仕様の確定度合によって大きな影響を受ける。本来ならば，仕様が確定してからコストを見積ればコストの精度は高くなるが，特に注文品については，仕様が確定する以前に見積らなければならないケースが多い。そして，この段階での見積りコストによって契約がなされるため，仕様の確定度合によるコストへの振幅を押えておくことが必要となる。

(iv) 仕様の追加，変更

仕様の追加，変更の状況ほど把握し難いものはない。しかも，設計コストに与える影響は大である。仕様の追加，変更の要因としては，ユーザからの要望，コンピュータ運用上の要望によるものがある。本来ならば，仕様を一旦確定した後は，仕様を凍結して開発を行うのが望ましいが，ユーザの実力，ユーザとの力関係等によってままならないケースが多い。しかし，製造に入ってからの仕様の追加，変更はさらに困難であるため，受け入れるならば設計の段階で受け入れるべきである。いずれにしろ，仕様の追加，変更を標準化するのは不可能であり，コスト算出にあたっては別枠で考えた方が良い。

なお，開発完了後の仕様の追加，変更は保守コストの中で取りあげる。

A2 製造コスト

製造コストは，他のコストにくらべ標準化が比較的容易である。コストの原単位としては，工数またはステップ数（ステートメント数）で考える。製造コストの変動要因としては，開発言語，コメント，テスト及びトライアル，仕様の追加，変更等がある。

(i) 開発言語

製造コストは，開発言語によって左右されるが，開発言語は他のコストを変動させる要因にくらべれば種類が少なく，標準化項目として考え易い。最も広く一般的に使われている言語を基準言語として設定し，他の言語に

ついては係数を定めればよい。あるソフトウェア会社の価格表によれば，1ステップ当りの価格比は，コボルを1とした場合次のように設定されている。

| | |
|---|---|
| コボル | 1 |
| アセンブラ | 1.25 |
| フォートラン | 1.5 |

(ii) コメント

コメントは，開発段階では意外に工数がかかるものであり，製造コストを高くする可能性がある。さらに，開発段階では，トライアル時点でようやく効果を発揮することになるため，軽視され易い。しかし，保守及び運用で短時間でトラブルを解決したり，適切な改造を行うためには，適切なコメントが非常な救いになることが多い。コメントは，現在は個人にまかされているケースが多いが，標準的なコメントを例示するだけでも，ある程度コメントの統一がとれる。標準化項目としては，コメントの表示方法（例えば＊印でコメントの周囲を囲む等）と若干のコメントのサンプルを決定すれば良いであろう。

(iii) テスト・トライアル

製造コストで捉えずらいのは，テスト・トライアルに関連する部分である。システムの複雑さ，トライアルデータの確保の状況，そして，担当者の個人差が大きくあらわれるからである。しかし，救いがあるのは，こういった問題はミクロで捉えようとすると難かしいが，マクロで捉えると案外当らずといえども遠からずといった値がはじきだせる。標準化項目としては，テスト・トライアルには製造コストの何割といった捉え方で充分ではあるまいか。

(iv) 仕様の追加，変更

製造段階での仕様の追加，変更は，設計段階の時以上に影響が大きく，仕様の凍結が望ましい。しかし，いろいろな状況で仕様の追加，変更は避

けられないであろう。標準化という面で見ると，設計段階で記したように極めて不確定な要因があるため，検討項目から外す方が良いと思う。

A3　保守コスト

ソフトウェアにとって，保守は必ずつきまとう問題である。注文品の場合は，開発段階でのコストの見積りには含めない方が良い。しかし，現在のソフトウェアは属人的な要素が強く，開発した人でなければ保守が極めて困難であるといったケースもあり，契約時に保守を含めて検討するのが一般化してきている。あるソフトウェア会社の例では，保守の費用は，1ステップ（ステートメント）当りでくらべると，言語によって多少の違いはあるが新規開発の場合の1.5倍となっている。これが，保守コストの標準化項目のひとつの方向を示していると思う。次に，流通品について見ると，流通品を販売する場合に，全く変更なしでそのまゝ適用できるケースはまず稀であろう。流通品本体をそれぞれの販売先のシステムに合わせて変更あるいは改造するのはもちろんであるが，周辺のソフトウェアの変更をともなうケースも多い。しかし，これもケースバイケースであり，契約上の問題でもあるため，標準化項目としては全く別枠として考えるべきであろう。方向としては，注文品の保守の場合と同様にとらえてよいのではあるまいか。

A4　その他

(i)　流通コストと流通品の販売数

流通コストを考える場合，販売数をいくつ見込むかは重要な問題である。需要と供給の関係で，ソフトウェアの種類によって販売の見込数が大きく変ってくる。現在，あまりこの問題が注目されていないのは，今流通しているソフトウェアが，注文品にしろ自社開発品にしろ，いずれも商品化をあまり意識せず，言いかえれば，販売数を1として開発したものであり，これを汎用化して流通品としたものが多く，真に最初から商品化をねらって開発したものの流通が少いからと考えられる。今後，流通品が増えるに従って，販売数の見込みによって類似のソフトウェアでも流通コストに大

きな差ができるため，販売数を無視することはできない。しかし，標準化という面では極めて捉え難いと思う。

さらに，販売数に派生する問題として，1社でソフトウェアを購入し，複数事業所で使用する場合，契約上の問題とも関連するが，コスト計算上での販売数のとり方の標準化も必要であろう。

(ii) 運用との接点

コストの範囲は極めてあいまいである。特に，運用との接点では，はっきり区切りをつけられない。基本的には，コストは極力単純な形で捉えられるようにしておくことが望ましい。特に，契約との関係では，次にあげる運用との接点は明確にしておくべきである。

＊教育（運用教育，ユーザ教育等）の範囲

＊トライアル

＊バグ等による事故の補償

A5 まとめ

ソフトウェアのコストは，ほとんどが人件費と見て良い。しかも，製造ラインの様に機械に制約されることもなく，人間の能力，意欲といった極めて尺度が設定しずらいものが原単位となっている。標準化が非常に粗いものになるのはやむを得ないであろう。

(i) 機密保護

機密保護の基本は，ソフトウェアを個人のものにしないことであり，そのための方法を標準化することが必要である。社会や企業という組織の中で，コンピュータシステムは，ネットワークとして完全に根をおろしつつあり，公共の福祉や安全に，深いかかわりあいを持つようになってきている。機密保護に関しては，このように公共に深いかかわりを持つソフトウェアが，その中味を全く一個人の担当に握られてしまうことが問題であり，そこに標準化の必要性があるものと思う。標準化項目としては，現在のところ，ウォークスルーの義務付けと監査が考えられる。

ソフトウェアの開発の一手法であるストラクチャードプログラミングは，ウォークスルーという。ソフトウェアの進捗管理には欠かせない手段，考え方を普及させてくれた。それまでは，ソフトウェアの進捗管理もまた，個人のものだったのである。そして，これは，ソフトウェアを個人から開放するという意味で，機密保護の上でもまた，おゝいに威力を発揮する。仕様書の確認段階はもちろんのこと，構造図のチェック，コーディング，テストとあらゆる要所を捉えて，ウォークスルーを行うことが有効である。ストラクチャードプログラミングの解説では，ウォークスルーに管理者が入ることは望ましくないとしているが，機密保護に関しては，管理者がウォークスルーに入るべきである。ソフトウェアの機密保護で最も重要なことは，管理者が機密ということにどれだけ関心を持ち，機密をどれだけ理解しているかに負うところが大である。現在のところ，考え方のみで具体性はないが，機密保護に関する標準化項目の狙い目は，ここに絞られるのではあるまいか。

(ii) 管理者

ソフトウェアの標準化で，直接標準化項目とはならないかもしれないが，重要な鍵を握っているのが管理者の問題である。

コンピュータシステムは，現在，ますます社会の中に浸透し，公共性を持ちつつあるが，ソフトウェアは，その性質上，生産性（コスト），品質，機密保護，引継ぎ等ほとんど全ての問題が，個人の能力とか意欲にゆだねられている。ソフトウェアは，それ自体の中に魔力を秘めており，それに携わる人を，その世界にのめり込ませ，終いには，ソフトウェアは完全に個人のものになってしまうという怖さがある。そこで，ソフトウェアを個人のものとしないために，標準化が必要であり，本当の意味で標準化の推進ができる管理者が重要な鍵を握ることになるのである。

生産性（コスト）について言えば，ソフトウェアの開発段階において，チームの管理者が，ソフトウェア開発の特質を知っているかどうか，人間の特性を理解し，うまく意欲を持ち続けて仕事をさせ得るかどうかによって大きく変ってくる。仕様の凍結の内容とタイミングによって，ソフトウェアの開

発が納期通りに納まるかどうか，しかも，ユーザが満足するものが完成するかどうかにもかかわってくる。さらには，ソフトウェアの開発を，運用を含めたトータルコストとして考えられるかどうかによっても，また，トラブル対策，進捗管理の巧拙によっても，生産性は大幅に変ってくる。

機密保護についても，ソフトウェアの機密保護は，現在は管理者に全面的に負っているといってもよいであろう。

引継ぎ，個人のプロモーション計画に基いて発生するソフトウェア開発途中での業務の引継ぎが発生するかどうか，引継ぎが発生した場合，タイミングと内容によって後の開発計画に支障のない引継ぎができるかどうか，ほとんどが管理者次第といってよい。

このように，現在のソフトウェアの問題に関しての鍵を握る管理者には，どんな資質が必要か，また，そういった資質を持った管理者が，現在のコンピュータ部門の中から育つかといった問題がある。

資質の中では，特に，人間に対して興味を持ち，理解ができるかということが最も重要であるということが，今まで述べてきたソフトウェアの特性からいって明らかであろう。問題は，こういった資質を持った人が，コンピュータ部門の中で育つかである。残念ながら，現状では極めて育ち難い要因が多く，育てる環境もできていない。現在，コンピュータ部門で仕事を希望する人を募れば，人間が嫌いでコンピュータとの付合いが好きな人が集る。それ以前に，社会構造が，人とのかかわりあいが非常に少なくて生活できる構造になっていることも見逃せない。生まれた時からテレビを相手に育ち，家族は少なく，近所付合いはない。受験，テレビゲームといった中で人間に揉まれずに来た人が，マイコンを経てコンピュータを業とすることを希望したとすれば，極めて問題の根は深い。情報処理技術者の国家試験にしても，現在のところあくまで技術者試験の域を出ていない。このように考えてきた場合，現在，ソフトウェアの標準化にとって，最も重要な課題のひとつは，管理者の問題であり，管理者の問題は，標準化の枠をはみ出して捉えねばならないところに難かしさがあると思う。

# Ⅳ　ＩＳＯ／ＴＣ９７／ＳＣ７の
# 幹事国の問題

## Ⅳ ISO／TC97／SC7の幹事国の問題

１９８１年１２月に開催されたＩＳＯ／ＴＣ９７の総会において，スウェーデンから，財政上の理由からＳＣ７の幹事国を辞退したいという申請があった。ＴＣ９７では，これについてあと1年間はスウェーデンが事務を継続し，その間に引き受ける国がなければ，ＳＣ７を消滅させるという議決を行った。

一方，Ⅱ2.2.3.1で述べたように，ソフトウェアの標準化に関してＳＣ７は多くの重要なitemを作成・審議中であり，かつⅡ2.2.2で論じたソフトウェアの標準化項目は，ＴＣ９７の中で審議するのが妥当であり，その場合ＳＣ７がそれを担当するのが筋である。

更に，今までは，ソフトウェアの中心的問題は，プログラム言語であったが，１９８０年代から２１世紀へかけては，除々にではあるが，学問的にも重心が2.2.2で述べたような問題に移行していくであろうことは明かであると考えられる。すなわち，ソフトウェア産業においては，標準化に関して，ＳＣ５で扱う項目よりも，ＳＣ７で扱う項目の方が一層重要性をもつことになるであろう。

以上の見地から，ＳＣ７をＩＳＯにおいて消滅させることには強く反対せざるを得ない。したがって他国から幹事国の申し出のない場合，我が国が引き受けることをソフトウェア業界，学会，通産省は真剣に検討する必要がある。

# V　参　考　資　料

V　参考資料

## 5.1　プログラム開発に関する新高度化計画詳細説明資料

ここでは2.1.2で前述した「プログラム開発に関する新高度化計画におけるソフトウェア（プログラム）の詳細を述べてあるが○印は標準化すべき項目，一下線部分は次のレベルで標準化が望まれる項目を示している。

### 5.1.1　制御プログラム

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| 1－(1) | 汎用コンピュータシステム用制御プログラム<br>（プログラムの内容）<br>単一コンピュータシステム又は複数のコンピュータを有機的に結合したシステムで、各種資源の最適配分を行い、システムの使用目的に良く適合し得る運用管理をし、使い易いインタフェースを利用者に提供することを目的とした制御プログラム（ミニコンピュータ、マイクロコンピュータ等の汎用制御プログラムを含む。） | ① ファームウエア化による処理効率の向上<br>② アドレス空間の拡大と大容量メモリの有効利用の促進<br>③ 階層化記憶制御及び一元的アクセス機能の性能向上<br>④ OSのモジュール化により簡単な手段で最適を選択を可能とする機能の拡充<br>⑤ 中央プロセッサ群と各種の周辺プロセッサ（フロントエンドプロセッサ、バックエンドプロセッサ、アレイプロセッサ等）による最適負荷配分等の最適資源割当て、スケジュール機能の性能向上<br>⑥ 制御プログラムの分散化<br>⑦ プロセッサ間通信とデータ転送の効率向上<br>⑧ ユーザインタフェースに対する標準化<br>⑨ システム運用・操作性の向上<br>⑩ 自動運転機能の性能向上<br>⑪ 信頼性及び保全性の向上<br>⑫ 機密保護機能の性能向上 | システムの大規模化、データベース化、TSS化、オンライン化が促進され、システム構成も多様化の傾向にある。<br>システムの性能、信頼性の向上はかなり進展しているが、今後も更に追求されるようになる。<br>ファームウエア化、専用プロセッサによるCPU負荷軽減等の性能改善が未だ十分でない。<br>システムの大型化に伴い、システム及びユーザのプログラム規模が拡大し、従来のアドレス空間を拡張する必要性が高くなっている。<br>また記憶装置に関しては、ハードウエア技術の進歩により、大容量化、高速化が進み、これらのソフトウエアサポートも実用化レベルに達している。今後はこれら大容量（低速）から小容量（高速）の記憶装置の中から用途に応じ適切なものが選択されるようにする階層記憶制御の開発が必要である。<br>コンピュータ要員の不足、人件費の高騰の傾向に伴い、システム運用、操作性の向上、更には自動運転の機能の強化が今後共必要である。<br>様々なコンピュータニーズに対応するため、各種複合コンピュータシステムが出現している。基本的な結合方式においては実用化の段階は終っているが、さらに新しい各種結合方式が出現 | 1－(1)<br>単一コンピュータ・システム用汎用制御プログラム<br>1－(2)<br>複合コンピュータ・システム用制御プログラム<br>1－(3)<br>リソース・シェアリング・コンピュータ・システム用制御プログラム | |
| 1－(2) | 専用コンピュータシステム用制御プログラム | | | 新　規 | |
| 1－(2)－イ | 周辺機器制御プログラム<br>（プログラムの内容）<br>多様な入出力機器を制御し、入出力情報の誤りの検出、形式変換などを行うプログラム | ① 自然語、図形、画像、音声等のパターン情報処理機能の基本機能の実用化<br>② ホスト計算機との機能分担・負荷分担の最適化<br>③ 接続基準の標準化<br>④ マンマシンインターフェイスの改善による操作性の向上<br>⑤ 処理効率の向上<br>⑥ 信頼性の向上<br>⑦ 入出力機能の仮想化 | 従来の周辺機器に比べ、ホスト計算機の前処理及び後処理に関して、かなり高度な機能を持つ周辺機器が出現し、それを制御するプログラムもますます高度化しているが、パターン情報処理機能については必要性が高まっているにもかかわらず、未だ実用化されていない。 | 新　規 | |
| 1－(2)－ロ | 並列処理コンピュータシステム用制御プログラム<br>（プログラムの内容）<br>1つの仕事を複数に分割し、複数のプロセッシングユニットを使って、効率的に並列処理を行うプログラム | ① 各CPU間の機能分担、負荷分担データ転送の最適化<br>② サポートプロセッサとの連結機能の向上<br>③ 処理効率の向上<br>④ 信頼性の向上 | レーダー処理等高速処理が必要とされるごく限られた分野においては実用化されているが、一般的システムにおける実用化は今後の課題である。 | 新　規 | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| 1－(2)－ハ | データベースマシン用制御プログラム<br>（プログラムの内容）<br>・データベース管理を専用に行うコンピュータを効率的に制御するプログラム | ① ホスト計算機とデータベースマシンとの機能分担、負荷分担の最適化<br>② データベースアクセスの効率向上<br>③ 複数のデータベースの制御<br>④ 信頼性の向上<br>⑤ データの独立性の強化・<br>（複数のデータモデルのサポート）<br>⑥ 既存データベースとの連続性の確保<br>⑦ 記憶階層構成の試作 | データベースマシン自体の方法論について数種類提唱されているものはあるが、決定版は未だ存在しておらず、試作研究段階にあると言える。<br>データベースの利用拡大によりホストマシンと最適な機能分担等を図るハードソフト一体となったデータベースマシンの開発が望まれる。 | 新　規 | |
| 1－(2)－ニ | 組込み型コンピュータ用制御プログラム<br>（プログラムの内容）<br>特定の機器、システムの制御を専用に行う組込み型コンピュータを制御するプログラム | ① 信頼性の向上<br>② マシン依存部分の分離による移植性の向上<br>③ ファームウエア化による性能向上<br>④ カスタム化の容易性と最適化 | 既に4ビット、8ビットのマイクロプロセッサの低廉化とともに大きく進歩してきたが、今後はアプリケーション分野の広がりと相まって16ビット、32ビットの高機能を備えたマイクロプロセッサの拡大が予想される。 | 新　規 | embedded の略 |
| 1－(2)－ホ | プログラミングワークステーション用制御プログラム<br>（プログラムの内容）<br>ソフトウエア開発・管理・文書作成等を専用に行うミニまたはマイクロコンピュータ用制御プログラム | ① マン・マシン・インターフェイスの改善による操作性の向上<br>② 自然語、図形処理等の基本機能の実用化 | 米国には、UNIX 等のシステムがあるが、我が国にはまだない。<br>ソフトウエアの開発、運用コストの増大から、これらコストを大巾に低減するワークステーションの開発が望まれる。 | 新　規 | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| 1－(3) | 仮想マシンシステム用制御プログラム<br>（プログラムの内容）<br>コンピュータシステムを、仮想化することによりソフトウエアの価値を長期間継続させるとともにソフトウエア開発、システム移行等を容易にすることを可能にする制御プログラム。 | ① 仮想ＣＰＵ制御、仮想デバイス制御、仮想通信制御機能の向上<br>② ファームウエア化による処理効率の向上<br>③ 操作性の向上<br>④ 仮想マシンの載合機能の向上<br>⑤ 異種ハードウエアのシミュレート機能の実用化 | バーチャルマシンは既にタイムシェアリング処理形態を中心として、利用されてきたがＯＳなどのレベルアップや移行に対してソフトウエアの長期命化を確保するにはまだ性能が低い。<br>バーチャルマシンの性能は、バーチャルマシン制御プログラムのファームウエア化等により、向上が図られているが、まだ十分とはいえず、適用分野の拡大のために、今後一層の高性能化が望まれている。 | 1－(4)<br>仮想マシンシステム用制御プログラム | |

## 5.1.2 通信制御プログラム

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| 2－(1) | コンピュータネットワーク用通信制御プログラム | | | | |
| 2－(1)－イ | ネットワークシステム用通信制御プログラム<br>（プログラムの内容）<br>コンピュータネットワークにおいて、コンピュータ間の通信をつかさどるプログラム | ① 標準プロトコルの採用<br>② 低位プロトコル処理効率、信頼性の向上<br>③ 高位プロトコル処理機能の整備と性能向上<br>○メッセージ・プロトコル処理<br>○ファイル転送プロトコル処理<br>○ジョブ転送プロトコル処理<br>○データベース・アクセス・プロトコル処理<br>○端末仮想化<br>○機密保護機能<br>○ネットワーク管理機能<br>④ プロトコル・サブセットの実現 | コンピュータ、端末及び通信網を有機的に体系化したネットワーク・アーキテクチャーが開発され、その内容の充実が進んだ。低位プロトコルについては、国際的にも標準化が進んでおり、性能向上がなされてきたが、高位プロトコルに関しては、要求が多様化しているため、必ずしも十分な性能向上が達成されていない部分がある。<br>ネットワーク・アーキテクチャーの標準化は、国際的に進んでおり、これを受けて我が国でもネットワーク・アーキテクチャーのプロトコルの提唱が行われており、これらに基づいたプログラムの試作がなされている。<br>今後は、国際的標準化動向を踏まえて、高位プロトコル処理機能に重点を置き、プログラムの実用化・性能向上を図ることが必要である。 | 2－(1)－イ<br>フロントエンド・プロセッサ用プログラム<br>8－(3)<br>コンピュータ・ネットワーク用通信制御プログラム | |
| 2－(1)－ロ | インテリジェント・ターミナル用通信制御プログラム<br>（プログラムの内容）<br>端末装置又は端末制御装置上で動作し、ホスト・コンピュータと | ① ローカル情報処理支援機能の充実<br>○日本語処理機能<br>○端末独自のユーザ・プログラム群を管理制御する機能<br>○プログラム自動作成機能<br>○リモート・ジョブ・エントリ支援機能<br>○タイム・シェアリング・システム用支援機能 | すでに各社で各種、各様のインテリジェント・ターミナルが開発され販売されているが、ローカル情報処理機能の充実、またマルチメディア端末用通信制御プログラムの実用化等が望まれている。 | 2－(1)－ロ<br>インテリジェントターミナル用プログラム | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| | の通信、端末制御又はローカル情報処理の支援を行うプログラム | ○センタ睡眠時の処理機能<br>② 保守性の向上<br>③ 音声、図形、画像等マルチメディア端末用通信制御プログラムの実用化<br>④ マンマシン・インタフェースの改善による操作性の向上<br>⑤ 通信機能の性能向上<br>⑥ 端末仮想化機能の向上<br>⑦ ROM化、ファームウエア化されたパッケージの実用化 | | | |
| 2－(1)－ハ | システム間接続用プログラム<br>（プログラムの内容）<br>ネットワーク・アーキテクチャの異なるネットワーク・システム相互間を接続するプログラム | ① プロトコル変換機能の処理効率、信頼性の向上<br>② プロトコル変換機能の汎用化<br>（対象となるシステムの拡大）<br>③ ネットワーク管理プロトコルの変換機能の性能向上 | 特定のネット・ワーク・アーキテクチャ相互間のプロトコル変換を行うプログラムについては、実用化されているが、汎用的なものはまだ開発されていない。 | 2－(2)－ロ<br>網間接続用プログラム | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| 2－(2) | コンピュータ・ネットワーク運転管理用プログラム<br>（プログラムの内容）<br>分散形のネットワーク・システムにおいて、ハードウエア、ソフトウエア、データベース等の資源を効率的に共用することを目的として、これらの資源を一元的に管理し、ネットワークの運転をつかさどるプログラム | ① ネットワークの運転機能の向上<br>o ネットワーク資源の最適化管理機能<br>o 自動運転機能<br>o 最適トラヒック制御機能<br>o ネットワークの運転監視機能<br>② 装置異状・トラヒック異状対処機能の性能向上<br>o エレメントの障害やトラヒック状態に応じた動的なルーチング機能<br>o MTTRの短縮<br>o 監視・検出・修復等の自動化率の向上<br>o ネットワーク障害分析、切り分けを容易にする諸機能<br>③ 遠隔保守機能の向上<br>④ サービス、資源の追加、削除、構成制御機能の性能向上 | 一応のコンピュータ・ネットワーク運転管理機能は実用化されているが、今後コンピュータ・ネットワークの更なる普及に伴い、運転管理機能の一層の向上や信頼性確保の観点からの装置異状・トラヒック異状対処機能の性能向上等が望まれている。 | 1－(3)<br>リソース・シェアリング・コンピュータ・システム用制御プログラム<br>2－(3)<br>コンピュータ・ネットワーク用通信制御プログラム | |
| 2－(3) | データ交換網用通信制御プログラム<br>（プログラムの内容）<br>電子交換機（回線交換用、パケット交換用マルチプレクサ等）又は交換用のノード・コンピュータ上で動作し通信回線（無線も含む） | ① 交換機能の性能向上<br>o 多様な端末との接続機能<br>o 処理効率<br>o 無線パケット交換機能<br>o 信頼性<br>② 付加機能の充実<br>o 蓄積機能（同報通信、メール・ボックス等）<br>o データグラム転送機能 | 回線交換機能及びパケット交換機能については、すでに実用化されているが、これらについて更に性能向上が望まれるとともに、蓄積交換の特徴を生かした更に高度な付加機能の実用化、性能向上、また網管理機能の充実が望まれる。 | 2－(2)－イ<br>データ交換用プログラム | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| | 相互の交換を行うプログラム | ③ 網管理機能の充実<br>o 柔軟なトラヒック制御機能<br>o 大容量トラヒック制御機能（複数リンク制御）<br>o トラヒックのモニタリング機能<br>o ルーチング<br>④ データ交換網の相互接続機能の向上<br>o プロトコル整合機能<br>⑤ 機密保護機能の性能向上<br>o 暗号化機能 | | | |
| 2－(4) | 高度通信制御プログラム | | | | |
| 2－(4)－イ | マルチメディア用通信制御プログラム<br>（プログラムの内容）<br>音声・図形・画像等の情報を従来の符号化した文字情報（アルファニューメリック、かな、漢字等）との整合化を図りながら、統一的に取り扱う通信制御プログラム | ① 文字、音声、図形、画像等のデータを体系的に扱うことのできる通信プロトコルの開発<br>② 多様な端末に対応する柔軟性の向上<br>③ 高速・広帯域伝送の信頼性高い制御機能の実用化<br>④ メディア間の相互変換機能の向上<br>⑤ データ圧縮機能の性能向上<br>⑥ 衛星通信、光通信に伴う制御機能の実用化<br>⑦ ネットワーク管理機能の向上<br>⑧ 機密保護、信頼性の確保 | 個々のメディアに対して実用化されているが、LSI技術及び音声、画像等の処理技術の進展に伴い、これらのメディアを統合して利用するマルチメディア通信の気運が高まっており、これらを実現するプログラムの実用化が望まれている。 | 新規 | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| 2－(3)－ハ | 構内通信網用通信制御プログラム<br>（プログラムの内容）<br>同一ビル、工場、大学、あるいは研究所などの構内で、電話端末、データ端末、交換機等を有機的に結合したローカル・ネットワークを制御するプログラム | ① 各種情報メディアの効率的通信機能の性能向上<br>o 効率の良い統一された伝送制御機能<br>o 伝送インタフェースのＬＳＩ化に伴う効率向上<br>o 通信管理機能（アクセス制御、誤り制御、優先制御、フロー制御、ルーチング等）<br>o トランスペアレンシイの確保<br>② ネットワーク管理機能の向上<br>③ 拡張性、信頼性、経済性の向上<br>④ 外部網との接続機能の向上<br>⑤ 光通信に対応した制御機能の向上<br>⑥ 国際標準に準拠したプロトコルの採用 | 個々の構内設置機器を制御するプログラムはすでに実用化されているが、構内情報処理の普及に伴い技術進歩を反映した操作性及び効率が良い構内通信網システムに対する需要が高まっている。 | 新　規 | |

## 5.1.3　言語プロセッサ

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| 3－(1) | 汎用言語プロセッサ<br>（プログラムの内容）<br>汎用高水準の言語プロセッサ | ① 国際標準の動向に準拠した言語仕様の設定、強化拡充<br>② 日本語処理、対話機能、ガイダンス機能の充実等による使い易さと柔軟性の向上<br>③ 処理効率の向上<br>④ 処理内容の高度化<br>⑤ 並列処理機能の充実<br>⑥ ハードウエア等の技術動向に対応した言語の実用化、性能向上<br>⑦ 新高性能言語の試作、実用化 | COBOL、FORTRANについては国際規格に追従したものが開発実用化されている。<br>またPASCAL等国際規格の存在していないものについても実用化されている。<br>汎用言語は国際的広がりを有するものであるため、言語仕様の設定は国際標準の動向を十分踏まえ、機能の拡充を図っていく必要がある。<br>一方言語プロセッサ自体としては、利用の容易化を図るとともに処理効率の向上等を引き続き図る必要がある。また言語の開発については、我が国ではこれまで皆無であるが必要なものは今後試作実用化するとともに、その普及を図ることが望まれている。 | 3－(1)－イ<br>汎用高水準言語プロセッサ<br>3－(1)－ロ<br>汎用会話型言語プロセッサ | |
| 3－(2) | 簡易言語プロセッサ<br>（プログラムの内容）<br>表形式記述、パラメータ記述、コマンド入力等の簡易な手段により、ＴＳＳやインテリジェントターミナルを利用したエンドユーザや専門のプログラマでない利用者に容易にアプリケーション・プログラムの作成を可能と | ① 入力記述の容易化<br>② 対話機能及びガイダンス機能の充実<br>③ 日本語処理等使い易さと柔軟性の向上<br>④ 適用範囲の拡大<br>⑤ 言語の標準化<br>⑥ 処理効率の向上 | コンピュータ利用者層の拡大に伴い、対話形式等で簡単にプログラム作成が行える言語プロセッサに対するニーズが高まっている。<br>インテリジェントターミナル等における画面操作については、ひととおりの機能の実用化がなされているが、利用の拡大、利用者の増大、利用形態の多様化等に対しては機能、操作性ともに不十分であり、画面操作、画面編集、問合せ処理、データ検索、データ編集等の機能の向上が必要である。一方利用者の増大に対して個別システム毎にまちまちの仕様が存在することは利用拡大に当たっての障害となるので標準化 | 3－(1)－ハ<br>問題記述用言語プロセッサ<br>3－(1)－ニ<br>表形式言語プロセッサ<br>3－(1)－ホ<br>プログラム・ジェネレータ | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| | する言語プロセッサ | | を図る必要がある。 | | |
| 3－(3) | システム記述用言語プロセッサ<br>（プログラムの内容）<br>組込み型のシステム、ベーシックソフトウエア及びファームウエアを記述する言語プロセッサ | ⑪ 国際的な規格化の動向に準拠した言語プロセッサの開発<br>⑫ 言語の高水準化<br>⑬ 記述機能範囲の拡大<br>⑭ 生成プログラムの処理効率の向上<br>⑮ 使い易さと柔軟性の向上<br>⑯ 多重割込処理機能の高度化<br>⑰ 並列処理機能の充実<br>⑱ 処理内容の高度化<br>⑲ 対話機能の充実<br>⑳ 保守性の向上 | 組み込み型のシステム、特にリアルタイム制御システムなどの記述は従来主としてアセンブラにより行われてきたが、最近ではマクロアセンブラや高水準言語の利用が進みつつある。<br>今後クロスソフトウエアの整備により、こうした高水準への動きは強まるものと考えられるが、これら高水準言語は不統一な状態にある。欧米では高水準リアルタイム制御言語の標準化の動きがみられており、我が国としてもこれらの動きに対応しつつ標準化を推進する必要がある。<br>またベーシックソフトウエアあるいはファームウエアを記述する言語プロセッサについては、現在メーカ毎に自社用のシステム記述言語が開発され利用されている。<br>今後ソフトウエアの開発等をより効率的に行うためには、一層の性能向上を図りつつ言語の高水準化、使い易さと柔軟性の向上等をひきつづき行う必要がある。 | 3－(1)－ヘ<br>ベーシック・ソフトウエア記述用言語プロセッサ<br>3－(2)－ハ<br>マイクロプログラム用言語プロセッサ | |
| 3－(4)<br>3－(4)－イ | 問題向き言語プロセッサ<br>図形処理用言語プロセッサ | ① 形状モデル記述、曲面処理等の図形処理機能の試作<br>② 記述の簡易化、対話機能の向上等による使い易さと柔軟性の向上 | ひととおりの機能については実用化されているが、既存言語から呼出す方式が多く、操作性、性能等の面で向上が必要である。ハードウエアの多様化、応用範囲の拡大に伴い、さらに高度 | 3－(2)－イ<br>汎用図形処理用言語プロセッサ | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| | （プログラムの内容）<br>グラフィック・ディスプレイ等を用いた高度な図形処理のための言語プロセッサ | ③ 言語の基本部分の標準化<br>④ 図形データベースのサポート機能の向上<br>⑤ 3次元図形処理機能の向上 | な図形処理機能の実現が必要であり、また国際動向等も考慮して言語の標準化を図る必要がある。 | | |
| 3－(4)－ロ | 言語プロセッサジェネレータ<br>（プログラムの内容）<br>言語仕様とハードウエア関連仕様とを与えることにより、言語プロセッサを生成するジェネレータ。 | ① 言語仕様記述言語の対応できるアーキテクチャの拡大<br>② 言語仕様記述言語の対応言語の拡大<br>③ コンパイラの処理速度を早くするための言語プロセッサオプティマイザの試作 | 一部の限定された言語の分野での実用化はなされているが、全体的には研究レベルである。<br>言語仕様並びにジェネレータの試作を行うとともに、生成されたプロセッサの処理効率向上を行うためのオプティマイザの開発が必要である。 | 3－(1)－ト<br>言語プロセッサジェネレータ | |
| 3－(4)－ハ | 特定アプリケーションプログラムジェネレータ<br>（プログラムの内容）<br>業務手順等を簡易に記述することによって特定分野のアプリケーションプログラムを生成する専用のジェネレータ | ① 専用化によるアプリケーション適合度の向上<br>② 処理効率の向上<br>③ アプリケーションプログラムジェネレータの適用分野の拡大<br>④ 対話方式ジェネレート機能の拡充<br>⑤ 日本語データ処理機能の拡充<br>⑥ 記述の簡易化を含む使い易さと柔軟性の向上 | 特定用途でのみ、一部実用化がなされているが、性能、操作性等の面で不十分である。<br>またアプリケーションプログラムジェネレータの適用できる分野が限られていることによる利用の制約もあり、多種、多様なアプリケーションプログラムを利用者が安価かつ容易に利用できるようにするため、ひき続き実用化、性能向上を行う必要がある。 | 3－(2)－ニ<br>問題向言語プロセッサ | |

## 5.1.4 システム開発・運用支援プログラム

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧長期計画 | 備考 |
|---|---|---|---|---|---|
| 4 － (1) | システム開発管理支援プログラム | | | | |
| 4－(1)－イ | システム開発プロジェクト管理支援プログラム<br>（プログラムの内容）<br>システム開発に要するマンパワー、費金等の予測、システム開発プロジェクトの工程計画の作成及び工程管理を支援するプログラム | ⑩ 有効な予測モデルの実用化<br>⑪ 管理データ収集機能の実用化<br>③ 操作性の向上<br>④ 対話型機能の性能向上<br>⑤ 複数システムに対するプロジェクト管理機能の性能向上 | システム開発プロジェクト以外の用途には各種のプロジェクト管理プログラムが開発されそれらのうちの一部、例えばPERT的なものがシステム開発プロジェクトにも利用されているが、システム開発プロジェクトに用いるものとして十分な機能を有するものはなく、システム開発プロジェクトを専用に管理するプログラムを開発する必要がある。<br>一方システム開発については計測指標が確立されていないことによる管理の困難性も見られるところから早急に指標を確立する必要がある。 | 新規 | |
| 4－(1)－ロ | プログラム管理支援プログラム<br>（プログラムの内容）<br>システム開発過程において、必要とされるあるいは作成されるプログラム群を集中管理するプログラム | ⑥ 更新履歴などのメンテナンス情報管理機能の性能向上<br>⑦ 複数ユーザに対するプログラム集中管理機能の性能向上<br>⑧ 日本語処理機能等による使い易さと柔軟性の向上<br>⑨ 機密保護機能の向上 | ライブラリアン、PANVALETなど各種パッケージド、プログラムが使用されているが、日本語処理機能等によるさらに使い易さの向上が望まれる。 | 4－(1)<br>プログラム管理・メンテナンス用プログラム | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧長期計画 | 備考 |
|---|---|---|---|---|---|
| 4－(1)－ハ | ドキュメント管理支援プログラム<br>（プログラムの内容）<br>プログラムに関するドキュメントを集中管理するプログラム | ① ドキュメントファイルの集中管理（複数ユーザの同時使用、機密保護）機能の性能向上<br>② ドキュメント内容の柔軟な検索機能の性能向上<br>③ 日本文による管理機能の性能向上<br>④ プログラム変更と同期した、ドキュメントデータ等の管理機能の実用化 | 日本語処理技術、データベース技術等の進展に伴い、プログラムに付随するドキュメント、データ等の管理も進歩してきたが、今後とも使い易さの向上等が望まれる。 | 4－(1)<br>プログラム管理・メンテナンス用プログラム | |
| 4 － (2) | システム設計支援プログラム | | | | |
| 4－(2)－イ | システム設計要求定義支援プログラム<br>システムに関する要求及び諸条件の記述とその間の相互関係、優先順位等を指定することにより、要求間又は要求と条件の間の矛盾を診断するプログラム | ⑩ 診断機能の性能向上<br>⑨ 対話的に使用し得るインタラクティブ機能の向上<br>③ 日本語による分析、定義システムの実用化<br>④ 多様なアプリケーションに対処し得る汎用的機能の試作 | TRWのSREMに開発の例が見られるが、一般的には試作レベルにとどまっており、機能的に十分な水準のものは存在しない。 | 新規 | |
| 4－(2)－ロ | システム構成設計支援プログラム<br>要求仕様に基づきシステムの設計仕様 | ⑨ 対話的に使用し得るインタラクティブ機能の向上 | TRWのSREMなど特定の領域で一部利用されているものは存在するが汎用的なものはま | 新規 | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| | をチェックし、要求と設計されたシステム間の矛盾又はシステムの最適化に関する診断を行うとともに、対話型式による設計仕様作成を支援するプログラム | ② 日本語による記述機能の実用化<br>③ 設計仕様のモジュール化を支援する機能の試作<br>④ 専用化による機能向上<br>⑤ 多様なアプリケーションに対処し得る汎用的機能の試作<br>⑥ 分散システムにおけるネットワークリソースの最適配置設計等ネットワーク設計機能の性能向上<br>⑦ センターにおけるハードウェア、ファイル、ソフトウェアの設計機能の性能向上<br>⑧ 分散データベース設計機能の性能向上 | だなく、一般的には試作レベルにとどまっている。 | | |
| 4－(3) | ソフトウェア作成支援プログラム | | | | |
| 4－(3)－イ | ソフトウェア設計要求定義支援プログラム<br>（プログラムの内容）<br>プログラム設計仕様書に記述されたプログラムの機能仕様に対応する具体的な処理手続を理解し、矛盾を摘出しつつ、プログラムの階層的分解及びデータ管理、 | ① 処理手続自動生成機能の実用化<br>② 対話機能の充実等による操作性の向上<br>③ モデルのあいまいさの除去<br>④ ソフトウェア要求の実現可能性判定機能の実用化<br>⑤ 適用分野の拡大<br>⑥ 最適なモジュール分割機能の実用化<br>⑦ モジュール・インタフェース決定機能の実用化 | ＰＤＬなどに開発の例をみることができるが、実用化されたものは殆んどない。 | 4－(2)－ロ<br>ドキュメント作成用プログラム | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| | 入出力管理の体系的記述を支援するプログラム | ⑧ モジュール相関図の作成機能の実用化<br>⑨ 効率予測と最適化機能の試作 | | | |
| 4－(3)－ロ | プログラミング支援プログラム<br>（プログラムの内容）<br>オンライン対話型の開発環境において、プログラムのコーディング、編集等の支援及びデバックの定量化、効率化等の支援を行うプログラム | ① プログラム、モジュールのライブラリ管理及び検索機能の性能向上<br>② プログラム、モジュールの修正・合成のサポート・ツールの実用化・性能向上<br>③ テストデータジェネレート機能の性能向上<br>○ プログラムとリンクした有効な検査データ生成<br>○ プログラムの全ルートを検査できるデータ作成<br>④ プログラムとデータの静的・動的トレース機能の性能向上<br>⑤ プログラムモジュール単位でのテスト機能の性能向上<br>⑥ インタフェース等の情報の検索機能の向上<br>⑦ モジュール間結合テスト機能の向上<br>⑧ 実行時間の予測機能の向上<br>⑨ 特殊プログラム（非同期処理プログラムＩ／Ｏエラールーチン等）のテスト機能の性能向上<br>⑩ バックトラッキング機能の実用化<br>⑪ 操作性の向上 | プログラム・モジュールを結合する技術、プログラムの構造を解析し、分岐の様子などテストのための情報を提供するようなプログラムは我国においても精力的に研究開発が行われ、既に実用化された部分も存在するが、今後更に性能向上を図る必要がある。<br>特に、既存のプログラム・モジュールの再使用率を上げるため、これを支援するツールの開発が強く望まれている。 | 4－(4)－ロ<br>プログラム・テスタ<br>4－(4)－ハ<br>プログラミング・エイド・プログラム | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| 4 － (4) | システム検査・評価支援プログラム | | | | |
| 4－(4)－イ | ソフトウェア検査・評価支援プログラム<br>（プログラムの内容）<br>開発者以外の者が行う仕様書とプログラムの突き合わせ検査、プログラムの品質測定・評価を支援するプログラム | ❶ 多様なソフトウェアに利用できる客観的評価基準の設定<br>❷ 仕様書から検査基準、評価基準等を指摘する機能の実用化<br>❸ 検査基準、評価基準とプログラムとの突き合わせ機能の実用化<br>④ 性能予測プログラムの実用化<br>⑤ 検査完了ルートの定量的測定機能の実用化<br>⑥ デバック結果を考慮した有効な検査データの生成機能の実用化<br>⑦ 言語プロセッサ検査機能の向上<br>⑧ プロトコル検証機能の性能向上<br>○ 多様なプロトコルへの適用<br>○ 多様な検査環境のサポート<br>○ ハードウェアからの独立性の向上 | プログラマ以外の者が行う検査用のプログラムとしては通常デバック支援等のために開発されたものが利用されているのが現状であるが、ソフトウェアの流通促進等の観点から第三者が客観的にプログラムを検査評価することを支援するためのツールについてのニーズが高まっている。プログラムの効率評価については、一部実用化されたものとみることができるが、効率以外の評価基準について客観性の高い基準を設定することが困難な面も多く未だ実用化されるに至っていない。 | 4－(4)－イ<br>言語プロセッサ検証用プログラム<br>4－(4)－ロ<br>プログラム・テスタ<br>4－(6)－ロ<br>ソフトウェア性能評価プログラム | |
| 4－(4)－ロ | システム性能評価支援プログラム<br>（プログラムの内容）<br>ハードウェア、ソフトウェアを含むシステムの性能、効率の評価の支援を行うプログラム | ❾ 多様な定量的評価レポートの作成機能<br>❷ シミュレーションによる性能予測<br>③ 使い易さと柔軟性の向上<br>④ CPU、チャネル、メモリ、ソフトウェアモジュール等のリソース利用状況の把握機能の性能向上<br>⑤ ボトルネック部分の指摘機能及びシステムチューニング項の指摘機能の性能向上<br>⑥ トレース・データによるシミュレーション・モデルの作成サポート機能の性能向上 | 目的を絞ったものは若干実用化されているが、それらはシステム性能の一部を評価するだけであって、ボトルネック部分の指摘等多様な評価レポートを作成する機能を有するものはない。また、シミュレーションにより性能を予測したりする高度な機能を有するプログラムは未だ実用化されていない。 | 4－(6)－イ<br>コンピュータ・システム性能評価プログラム | |
| 4 － (5) | ソフトウェア保守支援プログラム | | | | |
| 4－(5)－イ | ソフトウェア保守作業支援プログラム<br>（プログラムの内容）<br>プログラムの解析支援及びプログラムの一部変更に伴う影響範囲の自動解析等を行うプログラム | ① プログラムの解析機能の性能向上<br>② 変更事項の影響箇所の自動解析機能の実用化<br>③ 操作性の向上<br>④ 変更結果検証機能の実用化 | ソフトウェア資産の蓄積に伴い必要性は急速に高まっているが、部分的なツールは一部存在するもののシステム化されたソフトウェアは存在せず早期の開発が望まれている。 | 4－(1)<br>プログラム管理・メンテナンス用プログラム | |
| 4－(5)－ロ | ドキュメント作成支援プログラム<br>（プログラムの内容）<br>保守作業に有効な仕様書、設計書、使用説明書等のドキュメントの作成支援を行うプログラム | ⑨ プログラムからの文書作成機能の性能向上<br>⑩ 日本語及び図形表示機能の実用化<br>⑪ ドキュメント編集機能の性能向上<br>⑫ 対話機能の充実等による操作性の向上<br>❸ 入出力インタフェース、使用ファイル及びその項目名の表示機能の性能向上 | ソースプログラムを解析してフローチャート、クロスリファレンス等を出力するプログラムなど利用分野が限定された形で実用化がなされているが、保守作業を行うに十分なドキュメントを作成する機能が不十分である。 | 4－(2)－ロ<br>ドキュメント作成用プログラム | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| 4 － (6) | システム運用支援プログラム<br>（プログラムの内容）<br>制御プログラムの運用機能に加えオペレーションの一層の自動化、簡易化、合理化を支援するプログラム | ⑬ 各種運用形態に応じた柔軟性の向上<br>⑫ 無人化オペレーション機能の性能向上<br>⑩ 負荷に応じた運転構成の決定機能の向上<br>④ オペレータ間い合わせ機能の性能向上<br>⑤ システム運転情報の監視・管理・分析機能の向上<br>⑥ 保守計画・運用計画・ハードウェア増設計画等の計画作成機能の性能向上 | 実現例はあるが、コンピュータ・システム運用の合理化を求める声は強く、様々な運用形態に応じた利便性の高いプログラムに対するニーズは高まっており、ユーザのオペレーションの実態に即した運用支援プログラムの開発が必要である。 | 4－(5)<br>オペレーション・サポート・プログラム | |
| 4 － (7) | プログラム変換支援プログラム<br>（プログラムの内容）<br>異システム間のプログラムの変換を行うプログラム | ⑰ 汎用高水準言語ソースレベルの変換率の向上<br>⑱ 汎用高水準言語に基づくデータ変換機能の性能向上<br>⑲ 変換検証機能の性能向上 | プログラム資産の有効利用を図るため、各メーカ毎に自社機種への変換プログラムは開発されているが、コンピュータユーザ層の拡大に伴いさらに汎用的な変換用プログラムが望まれている。 | 4－(3)－イ<br>プログラム変換用プログラム | |

## 5.1.5 データベース用プログラム

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| 5 － (1) | データベース管理システム用プログラム<br>（プログラムの内容）<br>複雑、大量のデータを有機的に取用して、共用化したデータベースを効率よく管理するプログラム | ⑩ 柔軟性及び処理効率の向上等アクセス手法の性能向上<br>⑪ リレーショナルデータベース管理の実用化<br>⑫ 分散データベース、巨大データベース管理の実用化<br>⑬ 機密保護等ファイル系の安全確保機能の性能向上<br>⑭ データベース用言語の性能向上<br>⑮ ユーザインタフェースの改善による操作性の向上<br>⑯ 国際的標準化動向およびハードウェア等における新技術の動向をふまえたＤＢＭＳの性能向上<br>⑰ 図形、画像、音声、日本語文章などの情報に関するデータベースの高度な取扱いを可能とするＤＢＭＳの実用化 | ＤＢＭＳ（Data Base Management System）の利用は最近とみに高まっており、国産ＤＢＭＳのほか、著名な輸入ＤＢＭＳが多数利用されている。この傾向は今後とも続くものと考えられ、かなりのアプリケーションプログラムが、ＤＢＭＳと関連を持って利用されるものと考えられる。<br>このようなＤＢＭＳ利用ニーズの高まりを反映して、ＯＳ側の機能拡充への要求も高まってきている。<br>一方取扱うデータについても、アプリケーションの高度化から様々な形態を取り扱うニーズが高まっているがこれらについては現状では困難である。<br>また新しい動きとしてデータベースの分散化や様々のＤＢＭＳの統合化を図ることにより、利用の高度化を図ろうとする動きもあり、そのための研究も活発に行われている。 | 1－(5)<br>データベース管理システム用制御プログラム<br>8－(2)－ロ<br>データベース用言語プロセッサ | |
| 5 － (2) | データベース運用支援プログラム<br>（プログラムの内容）<br>データベース管理システムを効率よく運 | ① データの発生、評価、利用等に関する論理的処理のための各種ユーティリティの整備<br>○ 自動インデクシング<br>○ 自動辞書作成<br>○ シソーラス自動作成 | ＤＢＭＳ利用の普及を図る観点から各種のユーティリティの整備や、マンマシンインタフェースの改善等が進んでいるが、さらにより容易にＤＢＭＳの利用が行なえるように使い易さのための各種支援プログラムが望まれている。 | 4－(3)－ロ<br>データ変換用プログラム | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する将来度化計画 | 備考 |
|---|---|---|---|---|---|
| | 用するための支援プログラム | ⑫ 異種データベース間のデータ変換等、データ構造の動的変化への対応機能の実用化<br>⑬ データベース作成支援機能の性能向上<br>④ データベース再編成機能の性能向上<br>⑤ データの評価機能の実用化<br>⑥ エンドユーザ言語の改善等による使い易さと柔軟性の向上 | | | |
| 5 － (3) | 情報検索用プログラム<br>（プログラムの内容）<br>大量の情報の中から必要とする情報を検索するプログラム | ⑲ 文章・図形などを含み効率の良い情報蓄積・検索システムの実用化<br>⑳ ガイダンス機能等、検索時のユーザ・サポート機能の性能向上<br>㉑ 蓄積情報の構造と検索言語仕様の標準化の推進<br>④ 検索後のデータの処理・加工・編集システムの性能向上 | 多種のシステムが実用化され、汎用化の段階に入っているが、文章・図形などを含んだシステムがこれからの問題である。 | 5－(2)－ハ<br>文献検索用プログラム | |

## 5.1.6 アプリケーション・プログラム

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する将来度化計画 | 備考 |
|---|---|---|---|---|---|
| 6 － (1) | 基礎共通的情報処理用プログラム | | | | |
| 6－(1)－イ | パターン情報処理用プログラム<br>（プログラムの内容）<br>音声、文字、画像等各種パターン情報の入出力を中心とする情報処理用プログラム | ㉒ 音声認識機能の向上<br>○ 不特定話者の単語認識機能の性能向上<br>○ 不特定話者連続単語認識機能の実用化<br>○ 認識できる単語数の拡大<br>㉓ 文字読取機能の向上<br>○ 書物読取機能の性能向上<br>○ 手書きかな文字認識機能の性能向上<br>○ 手書き漢字認識機能の実用化<br>㉔ 地図・図面・映像・その他画像の入出力、処理の向上<br>○ 地図・図面処理機能の性能向上<br>○ 印影・指紋・声紋処理機能の実用化<br>○ 動画処理機能の試作・実用化<br>○ 3次元物体認識機能の実用化<br>④ 画像情報におけるデータ圧縮・復元・管理機能の性能向上<br>⑤ スーパーコンピュータ利用による処理速度の向上 | 音声・文字・画像等パターン情報については大型プロジェクトの開発成果により、認識等のアルゴリズムについては確立され、一部のものについては実用に供されているが不特定多数の音声・手書文字複雑な図形等については認識機能が未だ十分とは言えず、今後の利用を図るために、これら機能のより一層の性能向上が望まれている。 | 5－(2)－イ<br>パターン情報処理用プログラム | |
| 6－(1)－ロ | 文書処理用プログラム<br>（プログラムの内容）<br>日本語、図表処理 | ㉕ 日本文・英文・図表を含めた総合的取扱機能の性能向上 | 日本文の処理プログラムに関してはここ数年目覚ましい進歩を遂げ、実用に供されているが、 | 5－(2)－ロ<br>日本文処理用プ | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| | を含む文書処理用プログラム | ⑩ マンマシン・インターフェースの性能向上<br>○ 日本語（漢字・かな）入出力の性能向上<br>○ 音声入出力機能の実用化<br>○ 画像処理機能（特に文書との混合処理）の実用化<br>③ 索引付け、誤り綴りの訂正等編集機能の性能向上<br>④ 文書管理機能の性能向上<br>⑤ 電子メール、メールボックス等の機能の実用化<br>⑥ 会議支援機能の向上<br>○ 会議スケジュール通知機能の実用化<br>○ 議事録自動作成機能の実用化 | 今後は、図表、英文等を含めた総合的文書処理・管理機能や索引付け、誤り訂正機能等、高度な機能を付加することにより、高度な文書処理を可能にするプログラムの開発が望まれている。 | ロ プログラム | |
| 6－(1)－ハ | 知識情報処理用プログラム<br>（プログラムの内容）<br>自然言語の意味の解釈・複雑な物体の認識・推論・問題解決・学習・連想・翻訳等人間の頭脳活動に近い知識情報処理を行うためのプログラム | ⑩ 自然語解釈機能の実用化<br>② 機械翻訳機能の実用化<br>③ 文脈解析によるキーワード自動付加機能の実用化<br>④ 複雑な物体の認識機能の実用化<br>⑤ 推論・学習・連想機能の実用化<br>⑥ 知識情報処理を効率的に記述できる言語の試作 | 自然語の文脈処理、複雑な物体の認識、推論・学習等知識情報処理ニーズは高まっているが、アルゴリズムは未だ研究段階にある。実現されたときの効果は極めて大きなものであり、可能なものから試作実用化を図っていくことが必要である。<br>また、これらと合わせて記述言語についても試作を行い将来に備えることが望まれている。 | 新　規 | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| 6－(1)－ニ | 機密保護用プログラム<br>（プログラムの内容）<br>通信データの機密あるいは蓄積されたデータの機密を保護するためのプログラム | ⑩ 標準的なデータ暗号化機能の実用化・性能向上<br>標準暗号系の採用<br>鍵管理方式の向上<br>② プライバシー保護機能の実用化 | オンライン情報処理の進展に伴い、通信回線を利用した情報の交換、送信、受授等の機会が増大するにつれ、送信情報の機密保護、あるいは蓄積されたデータの機密保護の要請が高まりつつあり、暗号化等によるデータ・機密保護機能が重要性を増している。<br>米国においては、標準暗号方式（DES）、公開鍵暗号系などの検討が行われ、我が国でもその重要性に鑑み、これらとともに、これらを混合した方式等について検討が行われている。 | 新　規 | |
| 6－(2) | 社会公共用プログラム | ① 自動検診・治療機能の性能向上<br>② 遠隔診断機能の性能向上 | 医療事務や医療機器用のプログラムについてはかなりの性能向上が行われた。検診の精度向上、病院管理等検診治療プログラムについては、今後に期待するところが大きい。<br>地域医療システムについては実験は行われているが本格的普及はこれからである。<br>また医療機器は今後とも新機能を有するものが出現するものと予想され、これらを十分にサポートするプログラムの開発が望まれる。 | 5－(1)－ホ<br>医療用プログラム<br>5－(1)－ヘ<br>医療事務用プログラム | |
| 6－(2)－イ | 医療用プログラム<br>（プログラムの内容）<br>検診・治療の合理化促進、地域医療の増進や新しい医療機器の高度利用を図るためのプログラム | ③ 総合医療情報データベース機能の性能向上<br>④ 信頼性の向上<br>⑤ マンマシンインタフェースの性能向上<br>⑥ 地域医療情報の提供機能の性能向上<br>⑦ プライバシー保護機能の性能向上<br>⑧ 診療機能、医療事務処理機能、医療情報管理機能等を組み合わせたトータルシステムの実用化 | | | |
| 6－(2)－ロ | 教育用プログラム<br>（プログラムの内容）<br>パーソナルコンピ | ① マイコン等安価なハードウェアとの一体化<br>② 漢字処理、図形処理機能の性能向上 | 本格的な教育用プログラムは、教師の有する教育ノウハウのプログラム化に困難な面もあり | 5－(1)－ト<br>CAI用プログラム | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| | ュータ、視聴覚機器の活用等を可能ならしめる教育の合理化、教育機会の向上を図るためのプログラム | ③ 体系的な教育を行うための汎用的、標準的、学習プログラムの整備<br>④ ガイダンス機能等マンマシンインターフェースの向上 | あまり実用化されていない。<br>マイクロコンピュータを利用した簡易な教育機器については急速に発展する気配をみせており、これらを中心にして性能の良いプログラムの開発が進むものと考えられる。<br>本格的教育システムは、職業教育及び社会教育分野を中心として進むものと考えられるため、これらをサポートする性能の良いプログラムの開発が今後必要である。 | | |
| 6－(2)－ハ | 環境管理用プログラム<br>（プログラムの内容）<br>環境データのモニタリング結果をリアルタイムに処理、蓄積し、その結果に基づく必要な管理ができるようにするためのプログラム | ① 環境情報及び汚染情報等を用いて予測モデルによりシミュレーションを行い、理想的な環境管理をサポートする機能の性能向上<br>② オンライン環境情報及び汚染情報の収集、整理、蓄積機能の性能向上<br>③ リアルタイム処理機能の向上<br>④ 信頼性の向上<br>⑤ データベース利用機能の向上 | 環境データを一定の基準と照合して警報するプログラムについては実用例は多数あるが、風景・風向、地形等複雑な条件のもとで、拡散方程式をリアルタイムで解きながら適切な指示を与えるものは実用化されていない。<br>予測モデルの精緻化と、これをリアルタイムで処理する機能の実用化・性能向上が今後望まれている。 | 5－(1)－ロ<br>環境管理用プログラム | |
| 6－(2)－ニ | 防災・防犯用プログラム<br>（プログラムの内容）<br>地震・火災等の災害や犯罪に備えて、その被害を最小にとどめ、あるいはその | ① 地域的に集中して防災・防犯情報を管理し、災害あるいは犯罪が発生した場合に避難のガイダンス連絡等を行う機能の性能向上<br>② 広域災害情報の蓄積と予測モデルによる最適避難経路シミュレーション機能の性能向上<br>③ リアルタイム機能の性能向上<br>④ 人間行動学・心理学等の成果の適用 | 高層ビル等隙られた区域では実用化されたものもあるが広域をカバーするものは、未だ実用化されていない。防災時の人間の挙動については不明な点も多々あるので、これらの要因を盛り込んだ広域にわたる防災プログラムの開発が必要である。 | 5－(1)－ハ<br>ビル街・商店街コンビナート等防災・防犯情報用プログラム | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| | 発生の防止のための監視、安全処理、最適避難経路のシミュレーション等を行うプログラム | ⑤ トータルシステム化<br>⑥ データベース利用機能の向上 | | | |
| 6－(2)－ホ | 生活情報サービス用プログラム<br>（プログラムの内容）<br>社会生活に直結した情報を提供するプログラム | ① 病院、学校、買物など生活ニーズに応じた種々の生活関連情報のサービス機能の実用化<br>② 地域全体の経済活動を中心とした情報（経済、産業、人口等）の管理及びサービス機能の実用化<br>③ プライバシー保護機能の実用化<br>④ 各種情報提供機能（役所情報、電子図書館等）の実用化<br>⑤ 統計・アンケート調査機能の実用化<br>⑥ 同一端末から各種システムにアクセスできるトータル・システムの実用化 | CAPTAINシステムや都市及び地域計画サポートシステムや地域統計情報システムなど一部実用化されたものもあるが、地域情報データ等を広くサービスするレベルには至っていない。 | 5－(1)－イ<br>生活・地域経済情報用プログラム<br>5－(1)－ニ<br>基礎生活物資管理情報用プログラム | |
| 6－(2)－ヘ | 道路交通管制システム用プログラム<br>（プログラムの内容）<br>道路交通における人車等の流れを円滑にするためのプログラム | ① リアルタイム機能の強化<br>② 信頼性の向上<br>③ トラヒックを把握して、管制官ないし、運転操縦者にデータを表示する機能の向上<br>④ トラヒックの流れを最適化するためのルート変更、進入禁止等を行うトラヒック管理機 | 高速道路の交通管制については、トラフィック状況の表示等一部実用化されているが、一般道路におけるトラフィック把握機能、最適な信号制御機能等の実用化が望まれる。 | 新規 | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| | | 能の性能向上<br>⑤ 各種交通管制機能を全国的に統制するトータル・システムの実用化<br>⑥ マンマシンインタフェースの向上 | | | |
| 6－(2)－ト | 家庭用プログラム<br>（プログラムの内容）<br>家庭内で用いられるコンピュータのためのプログラム | ① ホームトコンピュータシステムの実用化<br>○ 防災・防犯機能の実用化<br>○ 省エネルギ機能（電源制御・ピークカット等）の実用化<br>○ 家事管理機能（キッチン・家計、電話番号等の備忘録等）の実用化<br>○ 余暇利用機能（娯楽、教育、健康等）の実用化<br>② ホームコンピュータシステムとネットワークとの接続機能の実用化<br>○ 生活関連情報データベースとの接続<br>○ 防災・防犯システムとの接続<br>○ マルチメディア網との接続<br>③ 家庭内就業機能の実用化 | マイクロコンピュータ、パーソナルコンピュータ等の低価格化に伴い、家庭の中にコンピュータが浸透しつつある。<br>現状では、ホビー的な使われ方が多いが、技術進歩に伴い、各種機器の制御、教育、社会情報システムの端末等に、多様な活用が期待されている。 | 新規 | |
| 6 － (3) | 産業用プログラム | ⑩ 組織内における異種システム間の情報の相互利用の高度化 | 企業内における計画管理のためのアプリケーションプログラムは、かなりの数が開発・利用されている。また分析・予測などのための個別のパッケージドプログラムも性能はかなり向上した。<br>企業内外の情報（データ）との総合的なつな | 6－(3)<br>経営計画・管理用プログラム | |
| 6－(3)－イ | 経営計画・経営管理用プログラム<br>（プログラムの内容）<br>財務計画、販売計 | ⑪ 業務の内容が定型化しつつある分野における共通的プログラムモジュールの抽出及び再利用を促進するためのモジュール間インターフェースの標準化 | | 6－(3)－イ<br>経営計画用プログラム | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| 6－(3)－イ | 画等経済活動を適切に展開していくうえで必要とされる各種の計画策定を支援するプログラム、人事管理し財務管理等組織を円滑に維持していくために必要な各種の管理業務をコンピュータ化するために必要とされるプログラム等経営計画・管理業務の高度化に資するプログラム | ⑫ 業務内容が定型化した分野におけるパッケージドプログラム化<br>④ マイコン、オフィスコンピュータ等安価なハードウェアの有効利用<br>⑤ 対話機能・図形表現機能などを含んだマンマシンインターフェースの性能向上等による意志決定支援機能の向上<br>⑥ 経営計画立案機能の性能向上・不連続的変化、不側事態等の発生に備えた機能的代替計画の策定機能の性能向上<br>⑦ 経営状況の把握、経営監査等の経営診断機能の性能向上、汎用化<br>⑧ 経理・会計処理、会計、財務データの蓄積、財務諸表の作成等の財務管理機能の性能向上、汎用化<br>⑨ 人事情報の管理、要員適正配置等の人事管理機能の性能向上・プライバシー保護機能の向上<br>⑩ ＯＰ手法、シミュレーション、数理計画法等の各種経営科学手法の処理効率及び拡張性の性能向上 | がりを意識したシステムは今後の課題となる。また共通モジュールとして性能の良く使い易いパッケージの開発がさらに望まれるとともに、オフィス環境にマッチしたマンマシンインタフェースの一層の改善が望まれる。 | 6－(3)－ロ<br>経営診断用プログラム<br>6－(3)－ハ<br>財務管理用プログラム<br>6－(3)－ニ<br>人事管理用プログラム<br>6－(3)－ホ<br>経営科学手法プログラム | |
| 6－(3)－ロ | 設計・生産用プログラム<br>（プログラムの内容）<br>コンピュータにより設計に必要な図面、 | ⑪ 設計の計画・実施・管理機能の性能向上<br>○ ＣＡＤ技術の応用<br>○ 技術データ、標準、規格、仕様データ等 | 設計計算については、各種のパッケージも実用化され性能向上がなされた。<br>今後はこの適用分野の拡大とともに計算値、 | 6－(5)<br>生産用プログラム | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| | 文字、数値情報を管理するとともに、設計に必要な数値計算を行ない、設計業務の合理化を図るためのプログラム<br>NC工作機械、産業ロボット等製造局面における各種の工作機械を制御するプログラム。<br>各工程における設備装置を有機的に結合して、工程全体の自動化、合理化を図るためのプログラム、各業種の業態に応じた生産活動を行うために必要な計画作成とそれに基く生産実績の管理を行うプログラム等設計・生産業務の高度化に資するプログラム | のデータベースとの接続（エンジニアリングデータベース）<br>○ 製造工程制御プログラムとの結合機能の性能向上<br>㉓ 製造工程制御機能の性能向上<br>○ 自動適応制御機能の性能向上<br>○ 工程間の総合的制御機能の性能向上<br>○ 設計支援プログラムとの結合機能向上<br>○ マンマシンインタフェースの改善による操作性の向上<br>○ 数値制御機能の性能向上<br>○ 自動・無人運転機能の性能向上<br>○ 3次元物体認識機能との融合<br>○ 信頼性の向上<br>○ 適用分野の拡大<br>㉔ 生産管理・品質管理機能の性能向上<br>○ 生産計画作成機能の向上<br>○ 生産活動において計画通りでない事態が発生した場合の処置を支援する機能の性能向上<br>○ 需要予測・生産計画との総合化<br>○ 製造工程制御プログラムとの総合化<br>○ 各工程の品質管理、検査機能の性能向上<br>○ 完成品検査機能の性能向上<br>㉕ 業務の内容が定型化しつつある分野における共通的プログラムモジュールの抽出及び再 | 図面、文字等設計に必要な情報を総合的に取り揃え、操作性のよいシステムの開発が望まれる。<br>工作機械制御プログラムについては、自動化の著しい進展に伴い、数多くのプログラムが開発されており、一層の制御機能の高度化と信頼性の向上が望まれる。<br>工場内の設備装置群を全体として効率良く管理するためのプログラムはかなりのものが開発実用化された。<br>生産工程の連続化、生産品種の多様化が続いており、大量の情報を迅速に処理する機能の一層の向上が望まれている。<br>生産管理プログラムについては各業種に対応して、それぞれかなり高水準のシステムが実用化されているが大部分は個別の適用に限られ、汎用性の高い使い易いモジュールの数は少ない。 | 5－(5)－イ<br>生産管理用プログラム<br>5－(5)－ロ<br>設計管理用プログラム<br>5－(5)－ハ<br>製造工程制御用プログラム<br>5－(6)－ハ<br>コンピュータエイデッドデザイン用 | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| | | 利用を促進するためのモジュール間インタフェースの標準化<br>⑤ 自動化機器、ミニコンピュータ等と中央コンピュータとの一体化<br>○ CAD／CAM／CATの有機的な結合 | | | |
| Ⅱ－(3)－ハ | 流通・サービス用プログラム<br>（プログラムの内容）<br>顧客・マーケットのニーズ、販売計画に従い製品の配送在庫を管理するプログラム、窓口受注業務の改善、サービスの提供、業務の合理化を図るためのプログラム。<br>CD、PoS、ATM（Automatic tellers machine）、自動販売機等窓口業務を代替する機器を制御するプログラム等の流通・サービス業務の高度化に資するプログラム | ⑥ 配送・在庫管理機能の性能向上<br>○ POSシステムの組込<br>○ 全国的な集中在庫、配送管理機能の性能向上<br>○ 自動倉庫管理機能の性能向上<br>○ 操作性の向上<br>⑦ モジュール間インターフェースの標準化<br>○ パッケージドプログラム化<br>○ 自動化機器等も含めリアルタイム機能の強化<br>○ イメージ情報（地図、手書き宛名）との組み合わせ<br>⑧ 窓口・受注業務支援機能向上<br>○ 顧客情報、商品情報等のデータベースとの接続機能の性能向上<br>○ 音声入出力機能等によるマンマシンインタフェースの改善<br>○ イメージ情報（印鑑照合等）とのリンク<br>○ セキュリティ、信頼性の向上 | 配送・在庫については、必要に応じた個別システムの開発が部分的につみ上げられており、一部には汎用パッケージらしいものも見受けられる。しかし、その適用率はそう高くないので、今後適用範囲の拡大が望まれる。<br>窓口受注業務については、オンライン端末の普及によりかなりの部分が合理化されてきており、各企業システムについては相当高度なものが利用されているが、企業間にまたがるものや、企業内の異システムを統合するものは今後更に性能向上を図る必要がある。<br>また業務機器の制御に関しては、業種、企業ごとに個別に対処して、かなりの性能向上、機能充実が図られている。今後ともマイクロプロセッサ技術の進歩や社会のニーズに応じて様々なサービス機能を有するものが実用化されるであろうが、これらについても信頼性の維持、操作性の向上等を図っていくことが必要である。 | 5－(4)<br>流通・サービス用プログラム<br>5－(4)－イ<br>配送・在庫管理用プログラム<br>5－(4)－ロ<br>保険・証券・銀行等窓口サービス業務用プログラム<br>5－(4)－ハ<br>販売管理用機器制御プログラム | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| | | ○ システム間の総合化<br>③ 業務機器制御機能の向上<br>○ 信頼性の向上<br>○ マンマシンインタフェースの向上<br>○ 多様なサービス機能の付加 | | | |
| 6 － (4) | 研究開発用プログラム | | | | |
| 6－(4)－イ | 実験支援・解析用プログラム<br>（プログラムの内容）<br>実験計画の立案、実験装置の制御、実験データの解析及び実験式の推定・検証を行うプログラム | ① 高速多量の実験データ収集、データ管理の実用化<br>② 実験プロセス制御機能の実用化<br>③ 実験データの処理機能（統計処理、回帰分析等）の実用化<br>④ 実験計画と実験スケジュールの総合化<br>⑤ マンマシンインターフェースの向上<br>⑥ パターン情報処理、日本文処理、データベース、翻訳等の利用機能の実用化 | かなりの実用化、性能向上がはかられ、パッケージドプログラムもある程度普及しているが総合化、操作性の向上が望まれる。 | 新規 | |
| 6－(4)－ロ | シミュレーション用プログラム<br>（プログラムの内容）<br>プロセスシミュレーション、ストラクチャアナリシス、有限要素法などの物理 | ① プロセスシミュレーション用プログラムの性能向上<br>② 構造解析用プログラムの性能向上<br>③ 非線形領域、流体モデル等への適用分野の拡大と性能向上 | ある領域では汎用プログラムがかなり普及しているが、一層の適用分野の拡大とマシン効率の向上が望まれる。 | 6－(6)－イ 構造解析用プログラム | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| | 現象シミュレーション用プログラム | ④ 計算効率の向上<br>⑤ マンマシンインタフェースの性能向上<br>⑥ パッケージドプログラム化<br>⑦ モデル作成支援機能の向上<br>⑧ スーパーコンピュータ利用による計算精度、速度等の性能向上 | | 5－(6)－ロ プロセス・シミュレーション用プログラム | |
| 6－(4)－ハ | 宇宙開発用プログラム<br>（プログラムの内容）<br>ロケットの打上制御、姿勢制御等の技術計算を行うためのプログラム | ① 信頼性の向上<br>② リアルタイム機能の強化<br>③ ロケット、人口衛星その他宇宙飛行体の運動のシミュレーションと最適化機能の性能向上<br>④ 振動・荷重・熱・電波干渉等の試験データ解析機能の性能向上<br>⑤ 打ち上げ、誘導、追跡データの実時間取得、解析機能の性能向上<br>⑥ 飛行安全のためのロケット落下域予測、被害確率解析機能の性能向上<br>⑦ 静止衛星管制・データ解析機能の性能向上<br>⑧ スーパーコンピュータ利用による計算精度・速度等の性能向上 | 宇宙開発の進展に伴って各種のプログラムが実用化されているが、本格的に宇宙開発時代を迎えて、大型ロケットの打上、誘導制御や高度な機能を有する人工衛星の制御プログラムの性能向上が今後とも必要である。 | 5－(6)－ホ 宇宙開発技術計算用プログラム | 6－(4)－ハ以降は、技術先端的な分野の技術計算用プログラムが挙げられている。この部分は、その発展のためにコンピュータ利用が必須で、またそのためにプログラムが十分に貢献できる分野である。また、この技術先端分野用プログラムの |
| 6－(4)－ニ | 原子力プログラム<br>（プログラムの内容）<br>原子力利用の安全性の解析・核反応制 | ① 信頼性の向上<br>② 燃料挙動、核的安定、熱伝導等の解析機能の性能向上 | かなりのものが実用に供されているが、原子炉の安全性の一層の追求が今後とも必要であり、炉の安全性解析や核反応制御の技術計算を中心 | 5－(6)－ニ 原子力技術計算用プログラム | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| | 御等の技術計算を行うためのプログラム | ③ 遮蔽、耐震解析等安全解析機能の性能向上<br>④ スーパーコンピュータ利用による計算精度・速度等の性能向上<br>⑤ データ入力（アナログ情報を含む）即時応答機能の高度化 | として一層の性能向上が求められている。 | | 成果が他方面のプログラムの高度化に十分寄与するものと思われる。 |
| 6－(4)－ホ | 資源探査用プログラム<br>（プログラムの内容）<br>震動解析、画像処理等資源探査に必要な解析を行うプログラム | ① 探査精度の向上<br>② 新しい探査手法に適したアルゴリズムのプログラム化<br>③ 大量データ処理機能の性能向上<br>○ スーパーコンピュータ利用による計算精度・速度の向上<br>④ 画像処理機能の向上 | 石油探査、鉱物資源探査にはかなり高度なプログラムが利用されているが、衛星による資源探査等リモートセンシング技術は今後ますます進展すると考えられるため、これらを十分にサポートする性能の良いプログラムの開発が必要である。 | 新規 | |
| 6－(4)－ヘ | 気象予測用プログラム<br>（プログラムの内容）<br>気象予測を適確に行うためのプログラム | ① 気象予測精度の性能向上<br>○ 気象衛星からのデータ処理機能の性能向上<br>○ 予測期間のバラエティ化<br>② スーパーコンピュータ利用による大量データ処理における計算精度・速度等の性能向上<br>③ 画像処理機能の向上 | 気象予測用のプログラムはかなり実用に供されているが、予測精度の向上や、解析時間の短縮化に対するニーズは高くこれらを可能にするためのプログラムの性能向上の必要がある。 | | |

| 分類番号 | プログラム名および内容 | 開発すべき事項 | これまでの開発状況 | 関連する旧高度化計画 | 備考 |
|---|---|---|---|---|---|
| 6－(4)－ト | 地震予知用プログラム<br>（プログラムの内容）<br>地殻の変化等各種観測データから地震の発生を予知するためのプログラム | ① 地震現象のモデル化による地震予知機能の実用化<br>② スーパーコンピュータ利用による計算精度・速度等の性能向上 | 試みられてはいるが、地震現象自体に対する理論も十分でなく、実用になるものはできていないが、周期からして、近い将来に大地震が起る可能性が多分にあり、地震発生の際の被害を小さくするためにも、このようなプログラムが早急に実用化されることが望まれている。 | 新規 | |
| 6－(4)－チ | 遺伝子工学用プログラム<br>（プログラムの内容）<br>遺伝子工学にかかわる技術計算を高精度かつ効率的に行うためのプログラム | ① X線照射パターン解析による3次元分子構造（DNA構造）解析機能の性能向上<br>② 遺伝子組替操作（インターフェロン製造等）の工程計画・管理機能の実用化<br>③ 遺伝子暗号解読機能の実用化<br>④ スーパーコンピュータ利用による計算精度・速度の向上<br>⑤ 無人長時間自動運転機能の向上 | 分子構造及びその機能を解析するためのプログラムは、相当数実用化されているが、解析に長時間を要するジョブが多いため、処理効率・速度の向上、無人運転化が望まれている。 | 新規 | |

―― 禁 無 断 転 載 ――

昭和５７年 ３ 月 発行

発行所　財団法人日本情報処理開発協会
東京都港区芝公園３－５－８
機械振興会館内
TEL（４３４）８２１１（代表）

印刷所　株式会社タケミ印刷
東京都千代田区神田司町２－１６
TEL（２５４）５８４０（代表）

[illegible]（持出厳禁）

| | |
|---|---|
| 受　付 No. | C-14 |
| 受付年月日 | |
| 作　成　課 | |