

オ五世氏とソフトウエア

1983年 3月 31日

新言語普及研究元-マ

目次

1. 第五世代におけるソフトウェアの開発とその環境

主査 斎藤信男

2. 汎用ソフトウェアの普及すべき条件

石井義興

3. 第五世代コンピュータソフトウェアについて

大村泰司

4. 第五世代コンピュータソフトウェア

岡田二郎

5. 設計への応用

志村 順

6. 第五世代計算機とソフトウェア — 新言語への期待と実態 —

玉井哲雄

7. 新世代コンピュータソフトウェア技術の普及と技術向上

吉村 敏夫

1. 第5世代におけるソフトウェアの開発と その環境

慶応理工学部

斎藤 信男

第5世代のコンピュータは、知的能力を飛躍的に増大したものと予想されている。このようなコンピュータの環境では、どのようなソフトウェアを作成、開発が行われるのだろうか。

現在のソフトウェアの開発は、1970年代に入ってから急速に発展したソフトウェア工学によって裏づけられている。そこでは、種々の技法や方法論が体系化され、また、それらを支援する道具類のコンピュータ上での実現が盛んに行なわれている。それらの目標は高くかかげられているが、その達成は必ずしも十分ではない。その原因には、いろいろのことが考えられる。1つは、技法や方法論の進歩が十分でなく、それらの形式的な取り扱いが困難であるということである。もう1つ

は、それらをサポートする道具や環境を実現する
ために十分な能力が、コンピュータに与えて
いることである。この両者の解決が、新世代
のコンピュータと人間の応用技術によってもな
らされる可能性は十分にある。

以下に、第5世代コンピュータにおけるソ
フトウェア開発の新しい方向や可能性に関す
る見解をまとめる。

(1) プログラム^の生成

ソフトウェアの構成要素であるプログラム
を作成することは、コンピュータの誕生以来、
多くの努力を向けられてきた問題である。プ
ログラムの設計や記述の仕方に関する
構造的プログラムミング^{の提案}、記述しやすいプログラム
ミング言語のいくつかの提案など、各方面の
努力が傾けられてきた。どのような方法論を
とるにしても、プログラムミング言語によ
て与えられる仕様を満たすプログラムを作成
することは、全く人間の能力に任されている。
これが、ソフトウェア開発の基本枠構であり、

人間の創造力の發揮でさる余地があるとともに、人間の持つ欠点もそこに負反映されてしまう。

この基本機構を変更するには大きく修正する一つの方法が、プログラムの自動生成という技術であらう。これは、目的とするプログラムの仕様からそれを計算するプログラムをコンピュータの力で生成するという方法である。仕様を記述するということは、本質的にプログラムを記述することと変わりはない^かが、それは抽象レベルの高い言語による記述であり、また原則としてプログラムが何をすべきかを決めるいわゆる要求仕様の記述である。

プログラムの仕様と書く言語は、いわゆるプログラムミドル言語の上位にあるということであらう。また言語ともなる。これにより、プログラムが行なうべきことと非手続き的に記述するのに対し、プログラムミドル言語はその要求を実現するアルゴリズムを手続き的に記述す

る。第五世代の中核となる PROLOG は、非手続き的にプログラムを記述するという特徴を
もっており、その意味では従来のプログラム
言語に比べて抽象度は高いと言える。実
際には、手続き的解決が必要になることもあ
り、そのための操作も必要になることもある
のが現実的要求であろう。PROLOG よりも更に
高階の言語を使って仕様を記述し、PROLOG
のプログラムを生成することもあり得る。

このような意味でプログラムの生成を考
えるとしても、第五世代コンピュータに期待で
さるのは、その高なる知的能力に基づいてプ
ログラムの生成という困難な問題を解決する
かもしれないことである。プログラム生成
の1つの方法として、プログラムの検証の過
程から必要となる操作の系列を導き出す技術
がある。第五世代コンピュータの推論機構が、
その場面では役に立つであろう。

(2) プログラムの再利用

ソフトウェアを、部品によって構成すると

いうモジュールプログラミングは、ソフトウェア工学の1つの技法として進取されてきたものである。機械工学や電気工学においては、その歴史が長いこともあり、部品の標準化、互換性の維持、流通性の向上などが既に確立されている。コンピュータのソフトウェアの開発においても、部品化とその再利用はどんどん推進されているのが現状である。

ソフトウェアの部品を管理するために、一般にモジュールベースと呼ぶデータベースが使われる。そこでは、必要なモジュールを陽に名前を指定して検索するだけでなく、その機能や構造の仕様から検索できることもできるなければならない。ここでも、仕様の記述という問題にぶつかる。

モジュールベースに属しているモジュールの再利用において、実際に問題となるのはモジュール間のインター^フエースの整合である。ソフトウェアの部品は、機械工学や電気工学における部品に比べて、どうしても自由度^度が多

くなくしてしまふ。そこで、モジュール間の結合~~を~~をとるということが困難な限り、そのインターフェースをとるモジュールがそれが必要にふつてしまふ。

このインターフェースは新たに作成しかなければならぬが、その仕様は確定してゐるはずである。(したがって、このインターフェースモジュールを自動合成することが考えられる。つまり古いプログラムであっても、その効果が大きいものこそ、コンピュータによる自動合成の対象としては望ましい。

(3) 人間機械系

人間とコンピュータとの有機的な結合と協同作業は、従来からその重要性を認められていたが、高度な知能をもつ第五世代コンピュータにおいては、よりすぐれた人間機械系の実現が可能となるであろう。現在の会話型システム(たとえば、Unix)には、未だ使いにくい点もあり、多くの問題点をかかっている。それでも、従来の一括処理に基づいたソ

ソフトウェア開発に比べて、人間の行かう作業
がより高度なり、~~と~~いえる。より~~高~~い
作業、力仕事的な作業は、なるべくコンピュ
ーで押しつけ、本当の意味での知的作業に
専念できる環境が得られるから、理想的な
ソフトウェア開発環境といえる。

D. Knuth による TeX と Web システムに
は、プログラミングの作業は、すべてそのド
キュメント・デモンストラーションの作成に、~~て~~置き換
える。勿論、ドキュメントの早見やすい形式
(たとえば活字形式)のプログラムが含まれ
る。ただし、コンピュータは、どんな形式のも
のであっても、文法的に正しい文字列であら
ばよい。このように考えて行かうプログラミ
ングを、Knuth は "Literature Programming"
と呼んでいる。すなわち、プログラムを作成
することは、1つの文学作品を作ることに匹
敵するものと考えている。現在の活字システ
ムは、必ずしも人間機械系としてすばらしい
わけではない。入力方式の改善、出力方式の改

良、辞書や模範の柔軟な参照など、人間機械系として多くの改善ができた点がある。そのような知的レベルの高い清書システムによつて、プログラミング作業も文学作品の作成と同じような知的レベルの高いものとなるであろう。

2. 汎用ソフトウェアが普及すべき条件

(特) ソフトウェア・エー・シー

石井 義明

以下汎用ソフトウェアが一般に普及するための諸条件について述べる。これは第5世代用ソフトウェアの普及のために特に問題となるものを述べたものではなく、現存するソフトウェアが普及した条件から抜き出されたものである。第5世代ソフトウェアが普及するためにも有用と信ずる。

1. パッケージ化されていること。

これはソフトウェアが単に存在するというだけではなく、そのソフトウェアがある概念的まとまりを持っており、多用途がそのまとまりの中で使用でき、ソフトウェアと使用層がフィットしてゐなければならぬ。

2. 簡単に導入できること。

そのソフトウェアが単に簡単にインストールできると言うだけではなく、そのソフトウェアを用いたユーザシステムが容易に本番稼働状態とならなければならぬ。

3. メリットが具体的であること。

ソフトウェアを導入した場合、「そのめずらしさ」によってのみ価値があるのではなく、実際のメリットを利用者が感じ、さらに利用を増大する方向で、利用者が拡大しなければならぬ。

4. 技術が外から見えないうこと。

ソフトウェアの価値を高める時、内部処理の優秀さを理解させなければならぬのではなく、利用者が、その利用のプロセスを通して自然に価値を認めなければならぬ。

5. 現在の問題点、困難点を具体的に解決し、利用者がその実感を持つること。(フーリングが重要)

6. そのソフトウェアを成長させ続ける母体があること。

ソフトウェアはその利用の多様化に伴って、成長しなければならぬ。(バージョン・アップし続ける)

この成長性を継続的に維持し続けることが普及をさらに増大させ、時代・需要にも適応させる基となる。

このためにはその作業を実施する母体が必要である。

7. そのソフトウェアについての賛同に答えられる母体があること。

6. と同様

8. 虫が出ても、他に同一の虫が出ないこと。

ソフトウェアに虫は立けられなく、互にどの虫が他のユーザに出ないようにすることが重要であり、それを実現するための方法と実施母体が必要である。

9. パフォーマンスが絶対に良いこと。

普及に際して、パフォーマンスは重要なファクターである。

パフォーマンスが悪いと普及の大きさを妨げとなる。

10. 利用者にとって新鮮さ、おもしろさがあること。

普及の始めの段階は、そのソフトウェアの新鮮さとおもしろさが不可欠である。(身体的な楽しみを与えること)

11. そのソフトウェアを利用して、定型業務用プログラムを作成した時は、時代の变化（OS、マシン、ターミナルの変化）があり、そのプログラムが長持ちすること。（ソフトウェアのいろいろなものからの独立性を保持すること）

12. パフォーマンス・チューニングが容易であること。
時代の变化により環境の変化に伴って、システム全体のパフォーマンス・チューニングは不可欠なものとなる。個々のシステムの最適化は全体の最適化とはならない。

以上述べたことは現存するソフトウェアの普及に伴い、特に重要なポイントを列記したものであるが、第5世代のソフトウェアが試作の後、本格的利用を前提として開発される場合は、配慮すべき点であると信ずる。

以上

3. 第五世代コンピュータと

ソフトウェアについて

日本電子計算株式会社

大村 泰司

新しいソフトウェア（論理型言語でPRのLQGをより自然言語に近づけたもの）の普及について、コンピュータ・ユーザーの1人として思いつくことを述べてみたい。

そのソフトウェアは、第五世代コンピュータの上で機能を発揮し、ソフトとハードは一体化されて、人工知能システムと言って差しつかえがなさう。したがってソフトウェアの普及と人工知能システムの普及の問題とを区別して話しを進めたい。

人工知能システムのマーケットニーズは、システムとして考えられるのに、EC&Tでも研究のとおり、自動翻訳、各種コンサルテーションシステム、色々のCAD分野への応用等がある。

勿論、上記以外にも人工知能的なシステム

として、我々が是非欲しいものの一例に次のようなものがある。

- ・ソフトウェア開発を容易にするためのツール
- ・ソフトウェアの検証精度を向上させるためのツール
- ・音声・手書文字等を入力として使えるシステム
- ・音声パターンによる音声識別システム
- ・画像処理・画像データベースを扱うシステム。
- ・データベースを自動作成、自動メンテナンスを行うシステム。
- ・あいまいな表現でデータベースの検索を行うシステム。
- ・コンピュータ・セキュリティへの対応システム。
- ・異機種・異プロトコル間のオンライン・ネットワークを構築するシステム。

これらに対するニーズは、現在もところ増え

さいていい。新システムに是非期待したい所である。

一方、従来から我々は、コンピュータを用いて、多くの情報処理を従来型手続言語、FORTH, COBOL, PL/I等を用いて行ってきた。現在では通常の数字と文字を扱う事務処理、科学技術計算は、そのほとんどがソフトウェア化されている。

しかし、なおかつコンピュータは、増加をやめず、ソフトウェア作成ニーズは、止まることを知らない。コンピュータ・ハード価格が安くなってきたことも大きな理由であるが、最大の理由は、当りまえのことであるが、情報処理ニーズの増加である。

しかも増加しつつある情報処理は、**“判断”**、**“推論”**を名む部分が多い。従来型情報処理の多くは、人手で行っていた計算データ・プロセスのうち、**“判断”**、**“推論”**およびその基である**“知識”**を欠いた部分を行っていた。単純な人手労働に代替から、インテリ

ジェンスを入れた人工頭腦的使い方への変革
と言えよう。こう言った従来型システムのレ
ベルアップ指向は、非常に高いものがある。

このように、新しいシステム、従来型シス
テムのレベルアップ双方に、身近な所に人工
知能システムの必要性は、出てきつつある。

しかしこれらのニーズは、現時点では処理
することが難しいこともあり、本当の意味
での需要ではない。人工知能システムへの需
要に変えていくための仕組み、もた大切であ
る。一般的には、新技術、新手法が、その技
術的可能性が高いにもかかわらず、普及しな
いという弊— というものも意外と多い。その
弊— が、あまりにも専門化、特殊化しすぎて
ユーザーに理解、使用しにくくなり、このた
めに、折角のすぐれた新技術が淘汰されても、
需要に届かずにつきに— という場合であ
る。

人工知能システムへのニーズを大きくし、
かつ、マーケット需要に育てるためには、マ

マーケットニーズを常に取り入れるための方法と、システムの成果をその都度公表し、マーケットの反応を迅速に捉えることができるような体制が、是非必要であろう。勿論、――

――方法論の検討も、必要であろう。

オ五世代コンピュータについては、シーズ
ミ等型であり、新技術の開発こそ最大の課題
であることは論を待たないが、プロジェクト
の成功のために普及につけてもより積極的な
対応を、お願いしたい。以上

4 第三世代コンピュータソフトウェア
(株)東洋紡報社
岡田=郎

昨今のハードウェア製造技術の進歩は目覚
ましいものがあり、^{最近の}マイクロ・コンピュータ
は数年前のミニ・コンピュータ^{1/4}の処理能
力を背に持っている。これに伴ってハー
ドウェア、コストは急速に低減してきた。

これに対しソフトウェア、コストは^{ソフト}ソ
フトウェアの複雑化、人件費の高騰に比例
増大する一方である。米国では既にソフト
ウェア、コストがハードのそれを上回り、こ
の格差は今後開く一方である。さらに
FARの調査によると、ソフトウェアは
受ける需要の増大により、製造に費
するコストが、ソフトウェアの1人当りの生産
コストも10%以上増加する見込みである。
以上の状況から、近年ソフトウェアの生
産性を上げるべく種々の努力がなわれてきて
いる。この過程でソフトウェア生産に於ける
問題点として以下の諸事項を確立する必要
がある。

(株)東洋紡報社 大阪支店 岡田=郎

性が指摘されている。

- ・ テスト可能か一貫した要求定義技術
- ・ ~~ソフトウェア~~ テスト技術 (テスト・ツール選定の基準や一定したテスト手順)
- ・ 保守の容易なソフトウェア開発技術
- ・ ソフトウェア信頼性の実際の計測法など
- ・ SE、プログラマーの作業の質や生産性を計測する基準

等。

これらの諸技術の確立には、ソフトウェア開発に使用されるプログラミング言語の性質によるところが大きく、上記の問題点を解決しようとする新しい高水準言語の確立が期待されている。述語論理言語について言えば、プログラムの形式に様記述で検証に述語論理が標準的に使用される場合が多いので、これらの問題の解決には非常に適していると言えよう。

さらに、プログラミングの対象範囲から近

年の動向を見ると、ハードウェアの低廉化に伴い、研究ベースでなくとも記憶容量の大幅増大を占有、並列処理や非決定動作等従来のコンピュータ使用感覚とは尺長と云われるプログラミングの許容されるようになってきた。^{のり} プログラミングの対象範囲も従来の数値処理だけでなく、記号処理を中心としたより人間の情報処理に近い解法に転がりつつある。これに伴い、知約なプログラミングを容易に記述しうる言語系に対する関心も急速に高まってきている。

述語論理型言語はエンド・ユーザー言語として直感的に用いるかどうかを別とすれば、知約情報処理—特に自然言語処理には非常に適した構造を有している。というのは、述語論理は自然言語の抽象化、形式化の研究に由来してあり、現在では自然言語よりもっとも近い形式的体系と云えるのである。

これに關係データベースの構造も述語論理を基にみるところから、知約データベース模

系システムの構築にも優れていると思われる。
以上の観点から、新世代 = パーオの
核言語として述語論理型言語が選定されるに
至った理由は、完全納得であることがあり、ソ
フトウェア企業としてこの核言語が次代の整
合言語として順調に発展することと強く期待
するものである。

ところで、論理型プログラミング言語、と
くに核言語のベースとなる Prolog の特長と
して

- ・ パターン・マッチング (= フィケー
ション) による手続きの呼び出しと実行
- ・ 再帰呼び出し可能
- ・ リスト処理機能
- ・ バックトラックによる非決定動作
等が扱われている。

Prolog によるプログラミングは、処理アル
ゴリズムを厳密に記述し、呼び出し、実行する。す
なわち型言語にいうところの記述して、
ループを並列的にモジュールとして記述で

こゝで、
データの内容に依存するものとして^{インプリメントに}表現される

・ プログラミングが明解で容易。

という長所があり、非決定性動作と相まって

自然言語処理、知識ベース・システム等処理

アルゴリズムを逐次的、確定的に記述しにくい

対象の記述には極めて適している。こゝで

が、プログラム記述の観点からすると、こゝで

の長所も、保守の側から見ると、プログラムの

順序構造が直線的に明示されていないので

プログラマー読みにくい。制御がデー

タフローに依存しているため、実態の信頼性を

保証しにくい等の問題がある。

また、現在の Prolog にはバックトラック

の可能性をキャンセルするための $\leftarrow$ 操作

の機能が乏しいが、この論理的整合性には後述の

点のこともこの点も関係している。たとえば

、論理的に正しいプログラムでも、バックトラ

ック時にエラーを発生する可能性があり、

$\leftarrow$ 操作は随所に記入しないとプログラムが

進行しないという話を聞く。この辺の論理的整理を行うことが必須であろう。

さらに、現在の Prolog のファイル・システムでは、複数のプログラマーが同時に同じファイルにアクセスすると、ファイル内容が壊れる危険性がある。この手の解決とファイル機能の拡張は、プログラマー一般に普及するために行う必要がある。以上の諸点の他に、核言語が実用言語として普及するためには以下の整備を望む。

- ・ ベースを拡張とする エンド・ユーザー言語の開発（データ・ベース問合せ言語、記憶記述言語、検証言語等）

- ・ 核言語より上記 エンド・ユーザー言語 < 記憶言語 とのインターフェース

- ・ 第四世代風マシンで稼動する核言語インタープリタ・コンパイラ

等。

5. 設計への応用

~~5844A 21日~~
構造計画研究所
志 不 可 11/12

私はかつて建築設計をした。積極的な
美しさを強調した建物を数多く設計して、世
界のスターになる予定であった。昔の仲間には^{とて}取っか
しくと言えないが、このレポートが目にふけ
る人たちはその道とは全く無縁なので心配
ない。あるとき私の人生を信じ曲げる人がい
て、以来情報処理の分野で生活している。

設計という行為は人にとってこの上なく
楽しい。~~最も~~最も和的と創造的な仕事
だからであろう。新世代コンピュータ・シ
ステムの目標はまさにここにある。“出来るも
のか。それはこんな楽しい仕事をコンピュ
タにやらせたらたまらない。”これは設計屋
の意見である。私自身はこの意見にはほのほ
いことである。今さら設計仲間には戻らない
し、そのうえ1990年代のコンピュ
ータ・システムにたてつけば、私の居場所がなくな
ってしまう。

設計分野向きのエキスパート・システムに
はどのようなことが期待できるのか。どこで

でニーズがあるのか。ニーズを具体的に表現
向或テーマに置き換えることが出来るのか。
これらについて考えてみたい。

1. 設計活動の要素

新世代コンピュータ・システムの性能はつ
ぎの3つに大別出来る¹⁾。

- (1) 問題解決・推論機能
- (2) 知識ベース管理機能
- (3) 知的インタフェース機能

問題を設計という分類に仕切って、それぞれの
の性能を具体的に示す¹⁾出してみたい(図-1)。

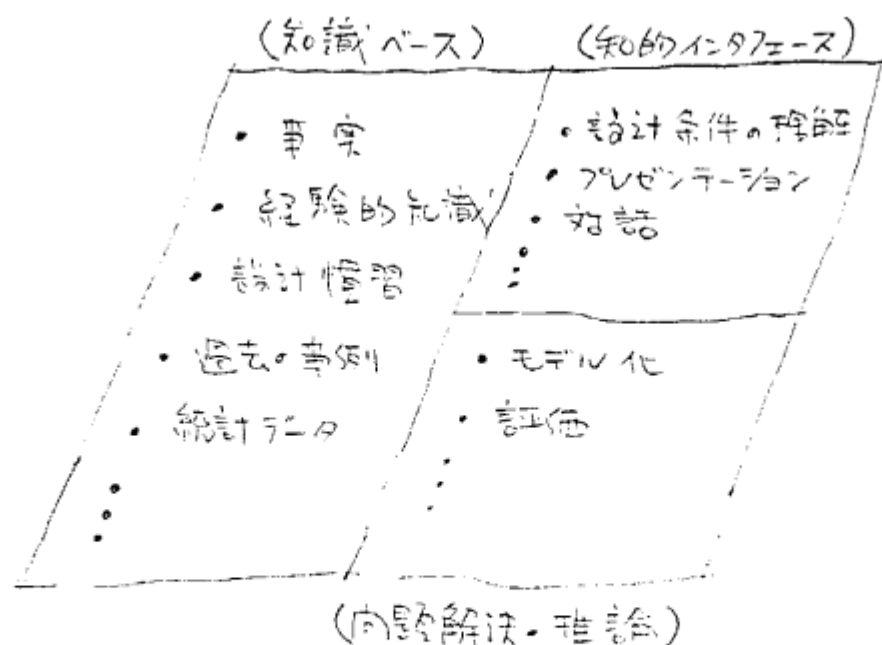


図-1 設計活動の要素

知識ベースには、教科書から得られるような事実に固まる知識や長い実務経験をもとにして得た知識がある。地震の多い日本と地震のないヨーロッパでは設計条件がちがう。このような設計慣習も知っていなければならぬ。過去の類似な設計事例や経験の積み重ねから得られる統計データ~~も~~も貴重な材料である。

知的インタフェースには、設計条件を容易に伝達する手段や設計結果の見やすい表現手段が必須である。設計のプロセスに特有な~~試行錯誤~~を実現する対話機能も重要である。

問題解決・推論には、モデル化と評価がある。現実是最も近いモデルを何をもよりどこに組み上げていくか。むずかしいであろう。評価にしても、機能面からの評価をはじめ、経済性、安全性、環境との整合性など限らない側面がある。

設計向きエキスパート・システムには、実現しなければならない課題が他にもあろう。

以上はとら合えず思いつくところである。こ
れらは設計現場での現実のニーズ^もである。シ
ステム化も試みつつある。しかし、今のコンピ
ュータ・システムの上には向かないとすると
限界がある。新世代コンピュータ技術の成果を
待ち望む次第である。

2. CAD/エキスパート・システム/人 間

T. Winogradの3世代論によれば
、number processorの性能
をもつコンピュータを第1世代といい、da
ta processor、knowled
ge processorの性能をもつもの
を第2世代と呼ぶのだという
のである。

マトリックス計算や応力解析など設計分野
で日常利用しているツール類はほとんど第1
世代コンピュータの上に実現した。図
形処理や画像処理^{用途の}ツールは第2世代コンピ
ュータ上にある。問題のエキスパート・シス

（この種のツールを統合化したのがCADであろう。）

の案理は、
「当然」の世代的コンコータ上である。

エキスポ・システムは、人間の設計活動
をどこまで取ってかわるゝとかでちるゝ
ある。1990年代には、図-1に示した
程度のことば実現できようである。何のとり
えもない貸しビルや首都高速道路程度はこの
システムでほとんど自動的に設計できよう。ロ
ーマ・オリンピックを記念して建設したイタ
リヤの見事な自動車道路や上野の西洋美術館
、五反田駅^{近くの}ホーテ化粧品ビルのちょっとした
アイデアなどは、このシステムをはるかに越
えたところにある。

人間の知的創造活動にはセンスのよしあし
がからむ。設計結果にもはっきり表われる。

→ 前者はセンスを全く無視した領域にあり、後
者^{の貸ビル}はセンスのある領域にある。配和の田島史をリ
ードした奥清源は、「素人という政治的な事は
何でも依頼できるが、センスだけは人に伝え
る方法がない。センスは人から^{しみ}とるもの
である。」といった。人が人に伝えられない

たは、センスがとれる、とてぬかぬ、階級の金でとる
あるとてぬかぬ。中身と外見と、仁義がまわっている、などいふのも
センスの表現である。

ことこコンピュータが認知おぼす^{がたい}くこれはSF
である。新世代コンピュータにも絶作に及ば
ない絶作である。西暦2000年になると、
T. Winogradがsensitivity processorの性能をもつコン
ピュータを第4世代と定義するかも知れない。
それまではこの辺の秘密は設計者、人間の楽し
みに残しておいしてほしい。

参考文献

- 1) 第5世代コンピュータ・システムの研究
(開発計画 概要、ICOT(57.8)
- 2) 第5世代プロジェクトの目標、ICOT

6. 第5世代計算機とソフトウェア

—新言語への期待と要望—

(社) 三菱総合研究所

玉井 哲雄

ソフトウェア産業内で日頃ソフトウェアの開発を行っている立場から、現在のソフトウェア開発の現状と問題点を簡単にみて、そこから第5世代計算機プロジェクトで生まれる新言語への期待することと述べる。またとくにPROLOGについて、使ってみて不便に感じることなどを述べる。

1. ソフトウェア開発の現状

1.1 開発手法

ソフトウェアの開発手法 ~~進歩~~ は、ソフトウェア・ハウス等の現場でも、決して速いとはいえないペースながら、徐々に進歩していることは確かだろう。少なくとも、ソフト

ウェアの開発には種々のツールを整備した方が便利だという認識は定着しつつあり、実際にはある程度のツールやプログラミング環境の導入・整備が、なされてきている。もっとも、いまだに大量のアセンブラーによるアプリケーション・プログラムをかかえているところもあり、レベルは様々である。

現在の大きな問題は、相変らず要求定義の分野であり、それも一時のような、形式的で厳密な要求定義技術や言語への期待はやや薄れ、ユーザーの要求にはどうしてもあいまいさが残り、変更は不可避であるとの認識が進んで、そこからプロトタイピングの提唱とか、ソフトウェア・ライフサイクル概念の有義論などが、でてきているように思われる。

プロトタイピング技法の1つの大きな目的は、動くプログラムをプロトタイプとして早期に作り、それを試用しながら要求の見直しと洗練を図るところにある。これは、まずきつ人とした要求を固めてから設計開発に入る

べきであるという、ライフサイクル概念で強調された哲学とは異なった考え方といえ、そういう文脈から、ライフサイクル有差論も出てくるわけである。

ソフトウェア開発の生産性向上のための究極的な手段は、要求仕様さえ定めればプログラムは自動的に作られるという自動プログラミングである。その意味で、ICOTにおけるプログラム自動合成の研究へ、期待するところは大きい。プログラム合成は我々も僅かながら研究を試みているが、難しい点が多い。論理プログラミングをベースとする新言語は、それ自身がプログラミングを大いに合理化するものであろうか。その上に構築される知識システムとして、プログラミングという高度に知能な作業がとりあてられるべき意義は、大きなものといえる。

1.2 ソフトウェアの検証

ソフトウェアのテスト・デバッグにかかる

コストは、開発費全体の50%を占めるとよく言われる。プログラムの正否性の検証技術は、現場ではまだ直接には役立たず、従来からのテストイングと、より体系的に、効果的に行うための方法論やツールの利用、プログラムの解析ツールの利用、形式的レビュー、テスト実績の測定、等がある程度使われ、少しずつ効果を出しているところといえよう。

新言語や新世代コンピュータは、いわゆる記号実行システムや正否性検証のシステムを作りやすくし、また性能を高めようだろうが、その時には、その対象となるべき系統言語は、死ぬ運命にあるのかもしれない。もちろん、論理プログラミングや関数型プログラミングにも、検証は必要であらうが。

2. 論理プログラミング

我々が実験的にPROLOGを使っている僅かな経験から、その便利な点と問題点を本題観のままあげてみる。これらより、PROLOG

に正しければ言わねえってことであらうか。

(1) 便利な点

(unification)

- ・ パターン・マッチング機能の便利さ。(構造をもったデータを、セクスメータを使わずにかけると $car(l)$ といわずに、 $x.l'$ などとすればよい — など)
- ・ 場合分けをそれぞれ別に書くことなどにより、アイディアをとりあえず動くプログラムにするのに便利。
- ・ 一般に、プログラムは短く簡潔になる。

(2) 不便な点

- ・ PROLOG では、ゴールが失敗した時に自動的にバックトラッキングが起こるか、実際の実用上は、失敗した時にそこを何かやりたいたいか、その時の状態を後で見たいということが多い。そのような操作が、全り自然に実現できない。
- ・ 論理上には正しいはずのものか、カットオペレータを適当な所に入れないと動かないということか、予想外に多い。

- ・ PROLOG の データ・ベースは 1 つだけで
共通のため、複数の人間によるプログラ
ミングや、大きなシステムの作成には不
便な点がある。また静的なデータに相当
するものは 2 つのデータ・ベースのみだが、
それを直接ユーザーが制御することは、
自然にはできない。
- ・ トップ・ダウンに考えてプログラミング
することはしやすいが、それを構造上に
反映できない。すなわち、階層構造やデ
ータの抽象化が直接には見えず、プログ
ラムが読みにくい。
- ・ 基本的なプログラミング技法とそれらの
利便性が欲しい。

3. その他

昨年 9 月に、プログラム自動合成、およびその関連で論理プログラミングにつき、数ヶ所の海外調査を行う機会を得た。自動合成関連では、USC/ISI や シュランベルグや研

究所など、論議プログラムミング関連では、エ
ディンバラ大学、ウプサラ大学、マルセイユ
大学などを訪れた。これらの大学や研究所で
日本の第5世代プロジェクトに関する局いの
は当然としても、一般に会った技術者や研究
者のほとんどが、第5世代について質問とし
たり意見を求めたりしたりしたのは、ちょっ
と驚いた。そのための広報努力が、たゞは
いえ、日本のプロジェクトでこれだけ世界か
ら注目をされていゝのもも珍しいだろう。

1983年3月25日

吉村氏

~~新言語普及委員会資料~~

新世代コンピュータにおけるソフトウェア技術の普及と技術移転

~~新言語普及グループ委員 古村鐵太郎 (管理工学研究所)~~

管理工学研究所

新世代コンピュータプロジェクトは極めて先進的な計画であり、多くの期待効果が既に議論の対象になっている。ここでは、ソフトウェア開発を専門とする者の立場から、重要とおもわれる点を幾つか挙げておきたい。

1 技術移転の対象

ソフトウェアハウスおよびシステムハウスの技術力向上と市場拡大の両面から、ソフトウェア開発技術とVLSI技術を取上げる。また、より基本的な見地からソフトウェア・エンジニアリング教育の進歩に期待する。

1.1 プログラミング技術の高度化

ここでの新世代技術の移転目的は少なくとも二つ考えられる。一つはプログラム開発技術の基礎的なし全面的な改善であり、もう一つはアプリケーション開発手段の質的向上である。

1.1.1 基礎的なプログラミング技術の改善

仕様の記述、プログラミング言語、検証技術等を有機的に総合したプログラミング技術(規範+ツール)を新世代マシンのうでで構築して、具体的な先導目標を実現する。ここに挙げたものは既に個別の技術としては数多く存在しているが、計算機アーキテクチャまでも含めて考えるプロジェクトのなかでの位置づけや在り方などを検討するのは恐らく今までにも例がなく、またこんごもその様な機会はその多くはないだろう。その意味でも今回はいわば新世代ソフトウェア・エンジニアリングを確立する絶好のチャンスである。

1.1.2 アプリケーション開発技術の高度化

現存するアプリケーションをもっと能率よく作ることも重要であるが、ここではむしろ今でも非常に作りにくい、非定型的な実務的プログラムをより経済的に開発することに新世代プロジェクトの成果、とくに人工知能技術の恩恵を期待したい。卑近な例としては、次のような対象を考える事が出来よう。

アプリケーションプログラムの中で各ユーザに関して個別に作らねばならない部分。(いわゆるカスタマイジング)

ハードウェアに依存するプログラム。例えばデバイスハンドラや、コードジェネレータなど。

1.2 新しいプログラミング教育体系

ハードウェアとソフトウェアとを先進的見地にたって白紙から作りあげるという新世代プロジェクトは、プログラミング教育の近代化にとっても又とない好機である。前記 1.1.1の内容にもとづく教材(ソフトウェアおよびハードウェア)の開発普及によって、いわゆる QWERTY syndrome から脱却した合理的基礎技術を持った次世代プログラマの出現

が期待できよう。

1.3 VLSI技術

単にソフトウェアを開発する事のみをソフトウェアハウスの仕事と限定して考える必要はない。ハードウェアの制約によってソフトウェアの機能をやむをえず限定する事はよく経験する。現在でもマイクロプロセッサ応用分野では、各種のバス規約に適合するCPUボードやメモリボード等が多数市販されており、これらを組合せて好みのシステムを作ることにはもはや常識だ。残念なことに今のところこのようなシステム技術をもっているソフトウェアハウスの数は極めて少ない。顧客の要求に対応し有利な市場把握をするためにはビットスライス・アーキテクチャの設計能力も近いうちに必要になるであろうし、さらにすんでやがて超LSIのアーキテクチャをソフトウェアハウス自身が設計する時代がやってくると思われる。VLSIの発達や標準バス製品の普及によって相当高能力の処理装置が比較的容易に実現出来るようになった現在、ソフトウェアハウスがソフトウェアの性能を最も良く生かすためのメインフレームを設計開発するのはむしろ当然ともいえよう。

ソフトウェア業がシステムハウスやメインフレームとの境界を乗り越えて、質的量的に成長するための一つの鍵がここにあると考えるとき、新世代プロジェクトで開発されるVLSI設計技術のソフトウェア業界への移転は極めて重要である。

2 技術移転および教育普及の方策

とりあえずad hocに幾つかの案を列挙してみる。

2.1 協同研究開発

前記 1.1で述べた各技術の(プロトタイプ)実現を、ICOTを中心とする研究機関、ソフトウェア業界および大学間での協同研究開発によって行う。

2.2 現実業務への応用

新世代技術(ハードウェアおよびソフトウェア)がある程度実用可能な水準に達したら、例えば国の業務を対象としたパイロット・プロジェクトをソフトウェア業に委託する。この場合今までまったくコンピュータ処理の対象にならなかった、あるいはなりえなかったものを選ぶことがより効果的であろう。

2.3 VLSI設計技術の公開

ソフトウェア向きVLSIの設計技術をソフトウェアハウスに公開する。また、VLSI設計のための開発環境(例えばDARPA もどきの)を実現し、これへのアクセスを奨励する。

2.4 教育および教材

新言語の教育に関しては専門プログラマの意識改革および潜在プログラマとしての幼年層に対する長期的な環境づくりに重点を置く。

2.4.1 テキスト

新言語や関連方法論の普及に役立つような、良いテキストを作る。媒体は印刷物のみに限らず、ビデオテープやビデオディスク等の視聴覚教材やCAIソフトウェアまでも含める。

2.4.2 ツール

現実のアプリケーションにおける新言語の効用を評価し、またこの言語の教育普及を促

進するため、次のようなハードウェアやソフトウェアを、教材として頒布する。

ソフトウェア： 既存の計算機システム（汎用機からパーソナル・コンピュータまでそれぞれいくつかの代表的な機種を選ぶ）上で稼働する処理系をおよび関連するソフトウェア環境

ハードウェア： 新言語マシンの為のchip, またはこれを実装した標準バス向けのボード、あるいは新言語マシンそのもの

2.4.3 教育プログラム

新言語は基本的には専門プログラマの為のツールである。従ってこの層に対する教育が最も重要である。比較的最近に情報工学科などを卒業した人々に対しては、直接新言語そのものの教育を行ってもさほどの障害はないだろうと思われる。一方既存のプログラミング言語により相当の実務経験を持つ層に対しては慎重な接近が必要だろう。この場合新言語への入門として、記号論理などを含むコンピュータ・サイエンスの基礎的な常識を教える事は大きな意味がある。同時に新言語によるソフトウェアと従来の言語によるそれとの間でのプログラムやデータの接合技術等をも提供する事を忘れてはならない。換言すれば技術的内容のみでなく雇用上あるいは業務上の心配や不安を引起こさないような方向づけが重要であろう。

以上