ICOT Technical Memorandum: TM-0128

TM-0128

知識情報処理用ワークステーションの研究開発
逐次型推論マシン：SIM の開発とその機能

内 田 俊 一

July, 1985

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

知識情報処理用ワークステーションの研究開発
逐次型推論マシン：ＳＩＭの開発とその機能

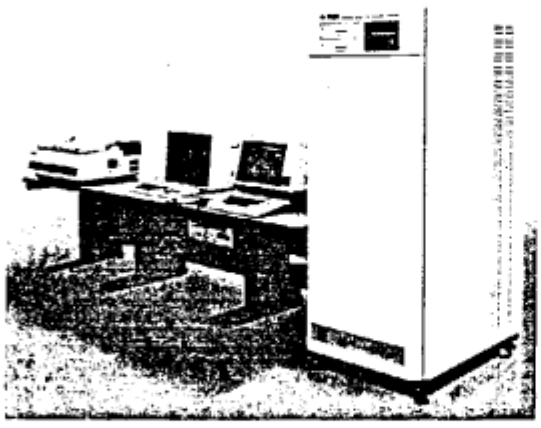
新世代コンピュータ技術開発機構
内　田　俊　一

1．はじめに

1982年4月からスタートした第5世代コンピュータプロジェクトも，その10年計画のうちの前期3年間を終え，先端的研究開発への本格的挑戦を意図した中期へと，新たな一歩を踏み出した。

前期の主要目標は，並列推論処理，知識ベース処理，自然言語処理等のソフトウェア，ハードウェアの要素技術開発と，中期より使用するワークステーションを始めとする研究開発ツールの開発であった。

特に，第5世代プロジェクトのためのツールの根幹をなす知識情報処理用のワークステーションの開発は，逐次型推論マシン（Sequential Inference Machine，SIM）プロジェクトと呼ばれ，新しい論理型言語の開発，マシンハードウェアの開発，さらに，プログラミングシステムとOSの開発を含み新しいワークステーションを概念設計レベルより作っていくものであった。



PSIシステム

SIMプロジェクトは，その中に，日本では，十分試みられたことのない，本格的パーソナルOSの開発を，それも，新しい論理型の言語で作るという試みが含まれていたことから，きわめて野心的なものときれた。

SIMプロジェクトは，前期3年内に，実用に耐え得るハードウェア，ソフトウェアを開発するという厳しい条件から，当初より，新世代コンピュータ技術開発機構（ICOT）研究所内に，専任のチームを作り，このチームの強力なイニシアチブのもとに，関連5社の研究者，技術者の協同開発チームを組織し，設計試作を進めた。

現在，当初の機能および性能目標を，達成することができ，SIMの基本部分であるPSI（Personal Sequential Inference Machine）と，そのOSであるSIMPOS（SIM Programming and Operating System）は，実用に供し得るレベルに至り，すでに，20数台のマシンが，ICOT内および，関連8社の研究開発グループに配布され，利用を開始している。この開発成功を受け，中期より，SIMPOSの機能拡充，ハードウェアの小型化，VANの拡充やDDX網との接続等第5世代プロジェクトの共通ツールにふさわしいものへと仕上げていく予定である。

本稿では，SIMプロジェクトの前期3年の足どりを追い，新しい言語やアーキテクチャ，さらには，新しいOS等，このプロジェクトで行われた，

本格的ワークステーション開発における種々の試みとこのようなワークステーションに求められる機能について述べる。

2．SIM システムの構成，機能，性能の概要

第5世代プロジェクトは，その始動に先立ち3年間の準備期間を持った。その間に，プロジェクトのハードウェア，ソフトウェアの成果物を，本プロジェクト関係者という狭い枠組を離れた広範囲の人々の共通財産とする方策として，将来を見越した新しい共通的プログラミング言語体系の確立と，それを実践するための共通ツールとしてのワークステーション（スーパーパーソナルコンピュータ）の開発・利用が決定された。

すなわち，プロジェクトの開始に先立ち，SIMと名付けられた，論理型言語に基いた高性能ワークステーションの開発が計画されていたわけで，この計画は，1982年6月にICOTが設立されると直ちに実行に移された。

1）SIMのソフトウェアシステム

ソフトウェアシステム（SIMPOS）の機能としては，まず，マルチウィンドウベースの対話機能を持ち，マンマシンインタフェースの優れたエディタ，ディバッガ，ライブラリアン，日本語処理プログラム等のプログラム作成ツールを備えた生産性の高いプログラミングシステム（PSと略す）をユーザに提供するとともに，OSの機能を，できる限りユーザに開放でき，ユーザが自分向きの一部カスタム化したプログラム開発環境を持てることを目ざした。

このシステムの想定するユーザは，知識情報処理の研究者であることから，当初から，かなり複雑で規模の大きなプログラム試作を担うことが考えられ，ディバッガやライブラリアン等では，ユーザが必要とするプログラムの実行途中の状態表示や，OSのプログラムモジュールの管理状況表示を丁寧に行うようにしたほか，OSの持つ各種のプログラムモジュールを，積極的にユーザに公開することとした。

このような狙いから，OSは，徹底的にオブジェクト指向型のモジュール化を行ったものとし，言語的にも，論理型の機械語KL0の上層に，システ

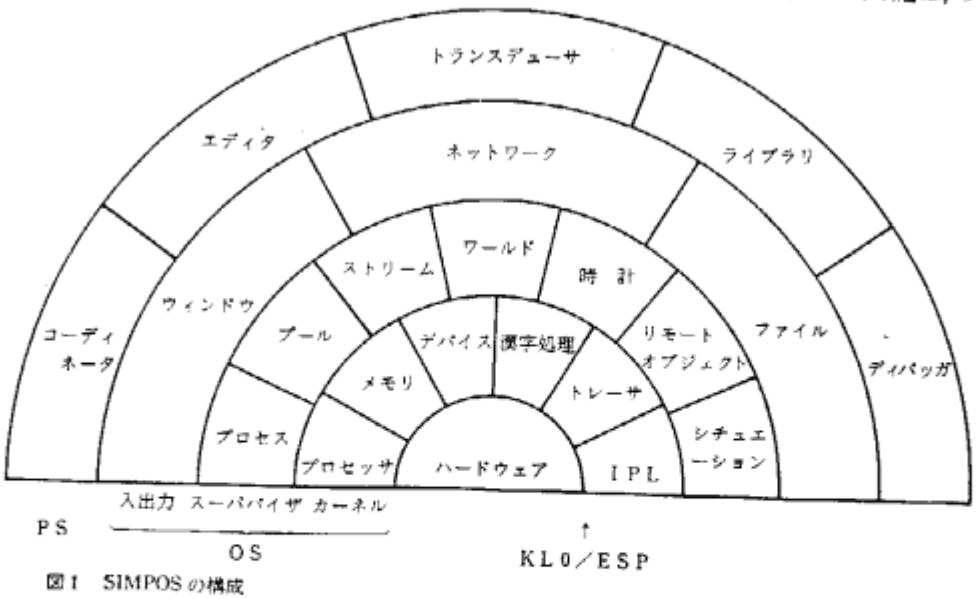


図1 SIMPOSの構成

SIMPOSは，プログラミングシステム部（PS），とオペレーティングシステム部（OS）に分けられ，それぞれ図のようなサブシステムから構成されている。クラス数は約700，ESPの行数が約12万行である。

ム記述言語ESP (Extended Self-contained Prolog) を設定した。ESPは，モジュール化機能として，クラスと多重継承機能を持ち，OSの持つモジュールを継承して用いることができるようになっている。また，ESPは強力なマクロ展開機能を有し，見やすく書きやすい言語となっている。

SIMPOSは，中期より，ツールとしての実用段階に入り，OS部分は，性能的な改良を行うとともに，ネットワークOSとしての機能拡張を行う計画であり，プログラミングシステム(PS)部分は，ウィンドウベースで作られたエディタ等のプログラムの対話性の向上や，エラー処理やヘルプ機能の拡充のほか，各種のユーティリティソフトウェアを付加していく計画である。

2）SIMのハードウェアシステム

ハードウェアシステムの機能としては，まず言語レベルとして，DEC-10 Prologのレベルを基準とし検討した。元来，DEC-10 Prologは，応用システム記述の観点から設計されており，マシン言語（機械語，KL0）として見た場合，ハードウェア制御やOS記述の点から，多くの機能拡張が必要であった。

そのほか，ハードウェアとしては，ソフトウェア開発に先立ち，試作を完了すべきこと等，開発期間に関する制約事項が，より厳しく存在した。

各種の要求事項を整理した結果，ハードウェアとしては，基本部と拡張部の二つに分けて作ることとした。基本部は，PSIと，PSI間を結合するLANを，拡張部は，PSIの後置型マシンとなる専用マシン，CHI(Cooperative High-performance Sequential Inference Machine，カイと読む）と図形等の専用入出力ワークステーションから成る。

PSIは，機能的には，DEC-10 Prologレベルの機械語を持ち，マルチプロセスをサポートし，入出力として，ビットマップディスプレイ，マウス等の対話デバイス，ディスク，プリンタ等の通常デバイスを持つ，スーパーパーソナルコンピュータとして設計された。

性能的には，DEC-2060上で走るDEC-10 Prologと，同等以上であること，メモリ容量的には，近い将来に渡って，不足を生じないよう

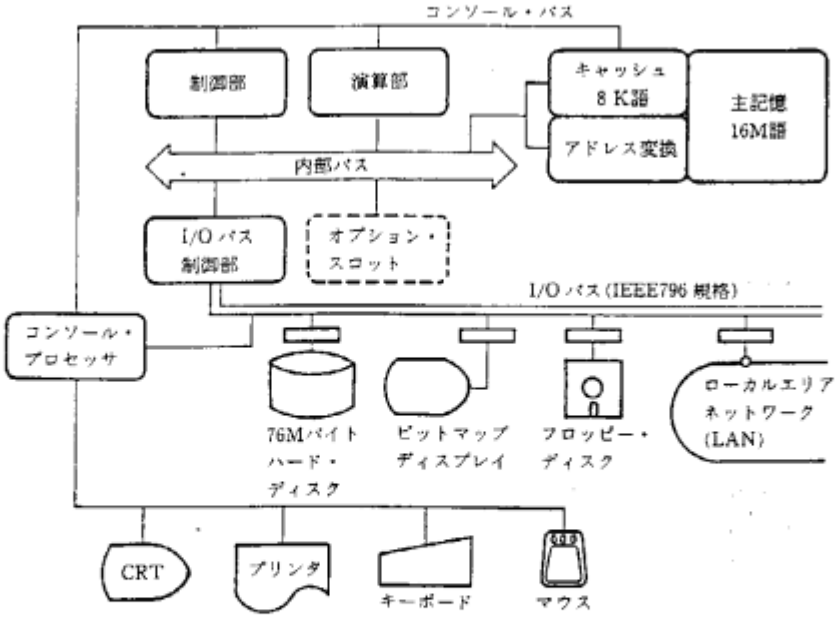


図2　PSIのシステム構成

16MW(80MB)まで付加できるものとした。特に，メモリ容量については，それまでのDEC-10 Prologのユーザがメモリ不足（DEC-10 Prologでは，ユーザ領域は，実質150 KW程度しかない）で，悩んでいたことから，特に多くすることとし，また実装に当ってもタイミングよく，256 Kbitメモリチップが利用できた。この大きなメモリが，ソフトウェア開発過程においても多くのメリットをもたらしている。

PSIのハードウェアの設計に当っては，当初より，早期に（1年半くらいで）ハードウェアを完成させ，さらに安定性，保守性も考慮した。アーキテクチャ的には，高級言語マシンでは標準的ともいえるタグアーキテクチャとマイクロプログラム制御を採用し，信頼性向上のためALUやレジスタ周辺にパリティチェック回路を，メモリには，エラー訂正回路を付加し，RASの向上も考慮した。ただし，論理型言語特有のユニフィケーションの実行やスタック制御には，いろいろと工夫を凝らした。

PSIでは，論理型の機械語KL0の実行は，ファームウェアで行っており，このファームウェアは，言語のインタプリートのほか，ガーベージコレクションを含むメモリ管理，プロセススウィッチングの制御，割込み処理など，ハードウェアとOSの間を結ぶ多くの処理機能を引受けている。このファームウェアは，約14 Kステップあり，この中には，開発者の多くの工夫が盛り込まれている。

PSIのファームウェアは，オブジェクト指向型の呼出し（メソッドコール）等のサポート，OSのプロセスの同期操作等のサポートを順次追加しつつある。このようなファーム化された機能は，今後再検討され，ハードウェア化へとすすむのが適当と思われる。

PSIについては，ハードウェア，ファームウェアの完成を踏まえて，小型化を行うとともに，中期より開始される並列推論マシンのソフトウェア研究開発のツールとして，マルチ化して使用する計画である。

SIMの拡張部の主要なものは，後置型の高速Prologマシン CHIである。CHIは，PSIの処理

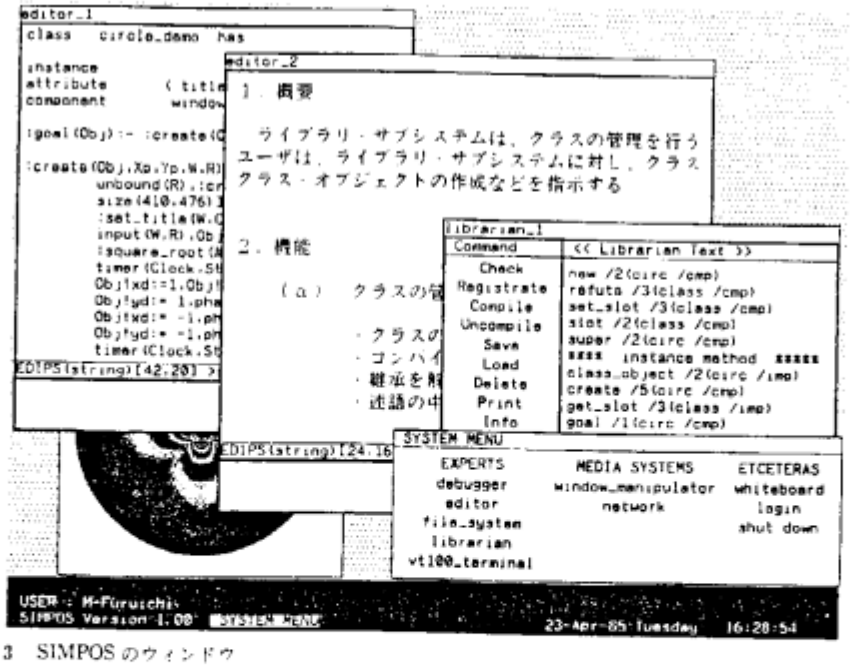


図3 SIMPOSのウィンドウ

いくつものテキストの編集，ディバッグ，ライブラリアンの操作等がウィンドウベースで行われる。SIMPOSでは，1文字は16ビットで表され，和文も英文と同様に扱える。入力はソフトウェアによるカナ漢字変換機能を用いて行う。

能力，メモリ容量を越える中～大規模の処理に対処することを目ざし，PSIの後置型マシンとして開発された。

その特徴は，PSIのような入出力制御やOSサポート機能は最小限とし，論理型言語の高速実行に焦点を合せた設計となっていることである。したがって，コンパイラによるプログラムの最適化がより徹底できるように機械語レベルをKL0より下げており，また，ハードウェア的にも，より専用化したアーキテクチャを用いている。また，高速のCML素子を用い，サイクルタイムはPSIの半分の100 nsとなっている。

これにより，平均的には，PSIの3～5倍の速度の実現が達成される。CHIの開発は，PSIと異り，最高速度の限界へ挑戦することに重点を置いており，研究的色彩の濃いものとなっている。現在，そのハードウェアとファームウェア中核部の試作が終り，その評価結果は，当初目標をほぼ満すものとなっている。CHIの開発においては，高速化のためPSIと異る言語の実行方式や命令形式を用いているものの，CHIは，PSIの機械語KL0の機能を包含しており，今後は，ESPと同様のオブジェクト指向型のモジュール化機能等も取り込んだシステム記述言語やPSIと接続して使用するためのソフトウェア等を開発し，ツールとして利用できるものへと改良，拡張していく計画である。

ハードウェアの拡張部は，このCHIのほか，図形，画像入出力および和文入出力を専用に行う高機能入出力サブシステムがあり，これらもPSIに接続され利用される。

3）SIMのネットワークシステム

ハードウェアの基本部は，PSIのほか，LANの開発を含んでいた。LANは，ETHERネットと同様の同軸ケーブルを用いた10 Mbpsの通信路と，LIA（LAN Interface Adaptor）と呼ばれる通信路と接続されるPSI等のコンピュータとの間を結ぶ高機能のコントローラから成っている。LIAは，PSI等1台ごとに1台ずつ用いられ，内部にマイクロコンピュータを含み，低レベルのプロトコル処理を，その内部で行い，PSI等の負荷を軽減している。

このLANを制御するソフトウェアは，SIMPOSのネットワークサブシステムとして位置付けられており，現在，プロセス間通信までが実装されている。

ネットワークシステムをOSの観点から見るとあるPSIから，他のPSI上のオブジェクトを操作するオブジェクトのリモート操作や，ネーム管理の実現方法等，分散OS的な機能が必要となり，これらの機能追加を現在すすめている。

ICOTのネットワークは，研究所内のLANと共に，ICOT外の諸研究グループの使用するPSI等もDDX網を経由して接続する構成となっている。現在は，ICOTで開発された関係データベースマシンDeltaやVAX等のマシンもハードウェア的には接続されており，そのソフトウェアの開発が計画されている。

3．SIMを通してみた知識情報処理用ワークステーションの持つべき機能

SIMは，第5世代プロジェクトの基本ツールとして十分な機能，性能を備えるべく，PSIやCHIのようなノードとなるプロセッサ，LANとそれをサポートする機能を持つSIMPOSを持ち，孤立したコンピュータではなく，面的な広がりを持ったシステムとなっている。

このようなシステム構成は，最近のワークステーションに共通のものである。SIMでは，このような共通のものの上に，さらに，知識情報処理を意識した機能を積上げている。

この中のいくつかは，すでに世に出ているLISPマシン等でも実装しているものであるが，SIMにおける位置付けも含め改めて紹介してみたい。

1）高レベルシステム記述言語のサポート

知識情報処理に限らないが，知的なシステムは

そのソフトウェアは作成において多くの試行錯誤的プロセスをくり返す。したがって，効率面は二の次とし，意図した機能を実現するプログラムが早急にできることが望ましい。ソフトウェア工学的にいうとRapid Prototypingが重要ということであろう。

Prologのような高レベル言語はこのような点ではきわめて望ましい性質を持つ。この種の言語では，メモリ管理等が，プログラマから隠されており，アーキテクチャの細部について意識することなくプログラムが作成できる点は，つまらない虫を防ぐ意味からも，有利である。LISPが好んで用いられるのも，この理由によるものであろう。

高レベルのシステム記述言語は，当然モジュール化機能をその言語仕様の中に含まねばならない。

SIMプロジェクトでは，論理型言語をベースとし，これに，クラスと多重継承機能を用いたオブジェクト指向型のモジュール化メカニズムを組合せたESPと呼ばれるシステム記述言語を設定した。SIMPOSは，約12万行の大きなシステムとなっているが，すべてESPで記述されている。

このESPで用いたモジュール化の方式は，ソフトウェアの生産性，保守性の点から見た効果がきわめて大きい。

これは，継承機能により，コーディング量を減らせたり，ソフトウェアのモジュール間インタフェースをモジュール内部仕様の細部と切り離して定義できることのほか，それまでに蓄積されているモジュール群が，システム記述言語の拡張，すなわち，ユーザ定義の，またはシステム定義の組込述語という形で見えることから，その利用がきわめて自然にできる。

特に，そのOSが同様の手法でモジュール化されている場合，プログラマは，OSが用いているクラスを継承して用いることができ，OS中に含まれるモジュールを，自分のプログラムの一部として容易に利用できる。

このようなオブジェクト指向型のソフトウェア構成法は，SmalltalkやLISPマシンのFlavor Systemで，すでに試みられたものである。しかし，Smalltalkでは，単一継承に機能を限定し，Flavor Systemでは，クラスと多重継承機能を，LISPマシンOSの上層部に追加した形をとり，特に，ウィンドウシステム周辺に部分的に用いている。

これらに対し，SIMPOSおよびそのシステム記述言語ESPでは，システム構築の基盤となっている論理型言語を拡張する形で，より徹底してオブジェクト指向型機能を導入している。この方式は，SIMPOSの開発自身に対してもソフトウェアの生産性，保守性の高さという多大な恩恵をもたらした。

この経験を通して，知識情報処理（人工知能含を含む）ソフトウェア開発においては，このオブジェクト指向型の機能は，不可欠のものとの認識ができた。この点から言えば，今後のこの種のワークステーションは，単にPrologやLISPをサポートするのみでは，不十分であり，オブジェクト指向機能の十分なサポートを同時に提供すべきと考えられる。

2）ユーザ（プログラマ）に対し開かれた構造を持つOSの存在

規模の大きなソフトウェアを作る場合，できる限り，既存のソフトウェアモジュールをそのまま利用したり，類似モジュールを編集して使うことが望ましい。（これは至極当然のこと。）

そして，OSは，このような材料となるモジュールの大倉庫である。ただし倉庫であるからには，ユーザが求めるもの（SIMPOSでは，求めるクラス定義）が容易に探せて取り出せるようになっていなければならない。

SIMPOSは，ライブラリ管理サブシステムが，システム定義クラスやユーザ定義クラスを一括管理しており，ライブラリアンと呼ばれるユーザインタフェースモジュールが，ライブラリ（倉庫と

対応）に関するサービスを行う。

ライブラリアンは，クラス定義のコンパイルやライブラリへの登録，削除から，クラス名やメソッド名によるクラスの検索等のサービスを行う。

また，ユーザは，自分の要求に合うようにOS機能の一部をカスタム化して取り込むことができる。たとえば，ウィンドウシステム中の必要なクラスを部品として利用し，自分の好みの機能，形状等を持ったウィンドウを作成できる。

3）上記1），2）に対する十分なファームウェア，ハードウェア的サポート

高レベル言語やオブジェクト指向型のモジュール化機能は，単純な実装を行った場合，実行速度の低下という形で，その“つけ”がまわってくる。

いくらソフトウェアの生産性が高いからといっても，その速度面での低下があまりに大きいと，やはり使われない。この点に関しては，コンパイラによる最適化等のみでは速度面での改善は十分ではなく，やはり，ファームウェアやハードウェア面でのサポートが必要となってくる。

PSIの場合は，ESPのモジュール化機能サポートのために，いくつかの機械語命令の追加と，これを効果的に用いるためのコンパイラの最適化等の工夫を行った。この意味から，PSIは，Prologマシンというより，ESPマシンと呼ぶ方がふさわしいものとなっている。

この種のアーキテクチャ上の工夫は，今後とも推進すべきものである。

4）マンマシンインタフェースのサポート

マウスやキーボードと組合されたマルチウィンドウシステムは，知的システムでは，きわめて有効な対話手段であり，その重要性は広く認識され，かつ，多くのシステムで実装されている。

しかし，ウィンドウシステムを用いた対話の形態は，エキスパートシステムのルールベースのエディタやディバッガ等の応用システムごとに，きわめて多岐にわたり，OS側が押し着せの形で提供する単純なインタフェースのみでは，機能的に不十分なものとなる。したがって，上記1），2）の機能を用いたプログラムにカスタム化を許すものとするのが望ましい。

これは，言語やOS，さらには，マシンのハードウェアレベルまで含めたサポートが必要なことを意味し，このような考えに基くサポートを行うことが，今後のワークステーションの条件となる。

6）プログラミング サポート ソフトウェア

この種のものとしては，エディタ，ディバッガ，インタプリタ，コンパイラ，ライブラリアン，ファイルやプリンタ関連のユーティリティなどがある。

知識情報処理のソフトウェアは，最も複雑な部類に属するソフトウェアであるから，このような直接的製作ツールも機能的に優れているとともに質の良いものでなければならない。

ここでいう“質の良さ”とは，使い勝手が良いということで，対話性に優れていること，コマンド体系に一貫性があること，応答速度が適当であることなどの基本的条件が満足されているとともに，欲しい情報が見やすくレイアウトされて呈示されること（エディタのプリティプリント機能，ディバッガのトレース出力など），欲しい情報を検索できること（ライブラリアンにおけるクラス名やメソッド名，継承情報の検索，ファイル中の必要ファイル名等の検索など）のようなそれぞれのソフトウェアモジュールごとに有効な機能が付加されていることが条件である。

このほか，OSの受けつけるコマンドの解析処理プログラムに，コマンドの一部省略形を許したり，コマンドのパラメータ（引数等含む）の形のガイド機能（ex. TOPS-20のCompletion機能），一連の操作を連続して実行したい場合コマンドをいくつか連続させて一続きの指令としてしまうコマンドのパイプ化（ex. UNIXのパイプ機能）などを組み込んであることも重要である。

また，ディバッガの実行途中で，その途中状態を保持し，エディタを走らせて一つのクラスの一

部を変更，コンパイルして，ロードした後，再びディバッガへ戻るような，途中状態の保持，復帰の機能が必要なほか，入力したコマンドのログをとる機能等もあると便利である。(これらは，まとめて，セッションマネージメント機能と呼ぶ)

ここで述べたような諸機能は，ソフトウェア開発一般に広く有効なものである。また，LISP マシンの OS 等では，このような機能の多くが，実装され，有効に機能している。

このような機能の実現に当って，そのワークステーションが，一つの言語体系のもとに組み上げられていることが，実質的に重要な条件となる。エディタやディバッガはもちろんのこと，OS のコマンドの文法等も，直接的，間接的に，そのワークステーションの基本言語の実行モデルや文法の特徴を反映し，それがユーザにとっては，一つの統一化された規範となり，使い易さに通じることとなる。この意味から，一つのワークステーションで，いくつもの体系の異る言語をサポートし，それぞれについて十分なプログラミング環境を提供することは，きわめて難しいと思われる。

以上述べた機能は，SIMPOS や PSI の開発を通して，その重要性を強く認識したものである。そして，SIM においては 1)，2)については，すでに開発に基本方針として取り込んでおり，3)～6)は，基本的には実現している。しかし，エラー処理やヘルプ機能，ユーザのためのユーティリティ等機能面でも実用面でもまだ不十分なものが多々ある。"使いやすさ"という尺度は，多くの使用経験をもとに，徐々に改良をすすめていかざるを得ない側面がある。

また機能の豊富さと処理速度等の性能は，一種のトレードオフの関係にあり，このバランス点をどこに求めるかも，重要な検討課題である。

4. おわりに

SIM プロジェクトを通してみた知識情報処理用ワークステーションの機能をソフトウェアの構成面に重点を置いて紹介した。SIMPOS は，数人の専門家を含む若い研究者集団，30～40人のチームが約 2 年間，勉強しながら作り上げたものである。"Prolog＋オブジェクト指向"の考えは，この若い集団には，似つかわしいものであったと思う。本稿では，少々，"オブジェクト指向"を強調しすぎたきらいもあるが，言語のベースが Prolog になっているのも，その効果を高めるもととなったと考えられる。

また，本稿では，PSI や CHI の性能，SIMPOS の構成等の詳細は割愛したが，それらは参考文献により補足していただきたい。

参考文献

[1] Daniel G. Babrow: IF PROLOG IS THE ANSWER,WHAT IS THE QUESTION?, *Proc. of FGCS* '84, pp. 138-145, Nov. 1984

[2] T. Chikayama: Unique Features of ESP, *Proc. of FGCS* '84, pp. 292-298, Nov. 1984

[3] K. Taki, *et al*: Hardware Implimentation of PSI, *Proc. of FGCS* '84, pp. 398-409, Nov. 1984

[4] M. Yokota, *et al*: A Microprogrammed Interpreter for PSI, *Proc. of FGCS* '84, pp. 410-418, Nov. 1984

[5] SIMPOS については，第29回情報処理学会全国大会，講演番号 4E-1 から 4E-10 まで，および第30回情報処理学会全国大会，講演番号 4E-1 から 4E-11 までを参照。

[6] CHI は，以前は，HPM と呼ばれており，これについては，第20回情報処理学会全国大会，講演番号 1C-4 から 1C-9 までを参照。