

ICOT Technical Memorandum: TM-0153

TM-0153

ICOT のワークステーション

上田 尚 純 (三菱電機)

内田 俊 一 (ICOT)

January, 1986

©1986, ICOT

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

ICOTのワーキング・ジョブ

— 逐次型推論マシンとIのアーキテクチャと基本ソフトウェア —

上田 尚純^{*1)}

内田 俊一^{*2)}

*1) 三菱電機 情報電子研究所

*2) 新世代コンピュータ技術開発機構 (ICOT)

1. はじめに

機械翻訳やエキスパート・システム等の知識情報処理分野の各種の処理を効効よく行なえる新しい計算機の開発を目指す5世代コンピュータ・プロジェクトが推進されている。このプロジェクトでは、基本となるプログラミング言語として、Prologに代表される述語論理型言語を採用している。論理型言語に基づくプログラミング開発を効率よく行なう研究開発ツールとして、ICOT（新世代コンピュータ技術開発機構）を中心に、本プロジェクトに参加している複数メーカーが協力して開発した知識情報処理ソフトウェア開発用ワークステーションが逐次型推論マシンP5I（Personal Sequential Inference Machine）である。

ウィンドウ・システムに代表される優れたマシン・マシン・インタフェースを提供するワークステーションは、プログラムの生産性の向上が期待できる道具として、米国を中心にその利用が急速に拡大しており、従来の汎用計算とTSSと端末との組合せによるプログラミング開発形態は、ワークステーションとLAN（Local Area Network）の組合せに移りつつある。人工知能分野では、プログラ

ラムの記述対象が複雑である事、目的とする性能と実現するためのアルゴリズムが明確でない場合が多い事等の理由により、研究開発におけるプログラム開発は試行錯誤の繰返しに頼る場合が多く、頻繁なプログラムの作り直しが生じる。このため、プログラムの記述時の豊富なライブラリ性能の提供や、マシン操作時の良好な対話性の提供は、効率よいプログラムの開発に不可欠と言える。また、プログラムの実行時にはこれら優れた性能の実現のためメモリが大量に必要となり、演算内容も記号の処理が中心となる。汎用計算機に代表される従来からの計算機は数値計算用に設計されているために記号処理の効率はいくらか低く言われており、またユーザが使用できるメモリ量も不十分である。大型汎用計算機にしても、十分な性能でウィンドウ・システム等の高度な対話性能を実現しようとするとの制限が多く、多数のユーザが知識処理プログラムを走らせるレスポンスが大幅に低下すると言われている。

(処理)
知識情報分野のプログラムの開発における以上の問題点を解決するには、現状ではワークステーションの利用が最適と言える。Lispマシンと呼ばれるLisp

用ワークステーションは米国において既に商用化され、普及してきている。

Prolog はLisp以上の記述能力を持ち、ソフトウェア作成効率の大幅な改善が期待でき、プログラムの実行時のハードウェアへの負荷も大さな事が予想され、

Prolog専用のワークステーションの開発はオーストラリア・プロジエクトの推進に不可欠と判断された。P&IはPrologによる実用規模の知識情報処理プログラムの開発、実行も可能なるように設計された、世界初のPrologワークステーションであり、実用を開始される段階まで開発が進んでいる。以下でP&Iのアーキテクチャと基本ソフトウェアについて説明する。

2. ハードウェア

2.1. システム構成

P&S I は Prolog プログラムの実行に最適化したハードウェア設計となっている。特徴として次のような点が挙げられる。

- ・ 最大 16M 語 (80M バイト) の主記憶容量
- ・ 論理型言語レベルの機械命令とファームウェア・インタプリタ
- ・ 16 ビットの文字表現コードの採用
- ・ 豊富な入出力装置

(知識情報処理においては)

P&S I のシステム構成を図 1 に示す。高度のプログラム記述能力と優れた対話性と実現するために多くのメモリと必要とする事、処理対象が複雑なためプログラム・サイズが大きくなり勝ちである等の事情があり、

4-(図 1)

P&S I では実用規模の ように入力 応用プログラムの開発も可能となる

よう、最大 16M 語の主記憶と実装で ように入力 ざる。更に、メモリ・アクセス高速化のため 4K 語 O2 セットのセット・アップシアティ、ブネ式による 8K 語の キャッシュ

と備えている。論理アドレス変換機構により1ページ(=1Kbit)単位での実メモリのプロセッサへの割付けが可能にしている。しかし、仮想記憶はサポートしていない。Prologでは、メモリ上の使用済み領域を回収するガベージ・コレクション(GC) (Lispと同じくメモリを断片的に消費して行き、メモリがなくなると呼ばれる操作)を行う。GCでは全メモリを調べるため、仮想記憶方式と比較するディスク・アクセスが生じ、処理に多大な時間が必要となる。P&Iでは実メモリを充分に用意して仮想記憶の必要性を減じる事で、この問題点を解消している。半導体メモリ素子の高集積化、低価格下は依然として急速であり、この解決策は今後も有力な選択の1つであろう。

P&IではPrologのソース・プログラムとほぼ1対1に対応する、高水準機械命令^(体系)を採用している。KLO (Kernel Language Version 0) と呼ぶ KLOでは200種を超える組込述語も提供している。これらには算術/論理演算、文字列やベクトルの操作、更には、OS等のシステム・プログラムを作成するのに必要となるプロセス管理、メモリ管理、拡張実行制御機能等が含まれる。KLOは

ファームウェアで作成したインタプリタにより解釈実行される。

次に、PSEIでの文字表現コードについて述べる。

。良好なマシン・マシン・インタフェースの提供のため、今日では日本語処理能力は不可欠である。PSEIでは、更に、機械翻訳を含む自然語処理への応用が想定され、英語、日本語だけでなく、外国の言語を扱える^(上巻160)が要求された。このため、PSEIでは16ビット・コードを採用し、カナ漢字の表現にはJIS 6226を使用している。

入出力インターフェースとしてマルチバス (IEEE 796) を採用している。入出力装置として図1に示すようなものがある。ビット・マップ・ディスプレイはコントローラ部に標準2Mバイト (最大6Mバイト) のイメージ・メモリを有し、また、マイクログプロセッサと専用ハードウェアも内蔵しており、グラフィックやイメージの処理を高速に行なえる。固定ディスプレイは200Mバイトのもので、最大8台 (1.6Gバイト) まで接続できる。LAN (Local Area Network) として、イーサネットと同様の形式に基いたものも用意している。

2.2 Prolog の高速実行機構

Prolog の高速実行のため、タグ・アドレスチャの採用、高速スタック機構、ファームウェアによるプロセッサ管理、メモリ管理のポルト等の対策を採っている。これらについて説明する。

メモリの1語は8ビットのタグ部と32ビットのデータ部とで構成される。

(図2) タグの内、先頭2ビットはGC形であり、残り6ビットで64種類の

データ・タイプを示す。データ・タイプとして整数、浮動小数点数、アトムと

い、たスカラ・データ、文字列、ベクタ、リストといった構造型データ、更に

はProlog の処理中にスタック中に生成される各種の中間データ等がある。Prolog

では変数(論理変数と呼ばれる)は構造体データを含めて任意のデータ・タイ

プを動的に持ちうるため、データの値とタイプは別にして保持しておく必要が

ある。タグ・アドレスチャによりこの要求を解決している。P&Eの内部バ

スや演算部もメモリの1語長に合わせて40ビット・アドレスチャとなっており、

タグ部とデータ部と並列に処理する。タグで示されるデータ・タイプに応じた

←(図2)

(プログラムの処理系は)

処理を、フレームウェアと一体化して、高速に行なう。

プログラムの論理空間は256ビットで表現され、 $2^{32} = 4G$ 語の単一論理空間を提示している。(図3) この論理空間は各々16M語を持つ256ヶのセグメント(エリアと呼ぶ)に分割される。エリア単位でヒープ領域あるいは

4-図3

スタック領域のいずれかに使用される。ヒープ領域はプログラム・コードや動作を持つ変数の値を置くのに使用され、プロセス間で共用できる。スタックはプロセス毎に4ヶ割り当てられる。ヒープ領域はシ

ステム用に最低1ヶ必要なため、 $(256-1)/4 = 63$ ヶのプロセスが並行して動作できる。

PSIは^(高速処理のため)プロセス管理やエリアの使用状況の^(の一部)管理をフレームウェアで行っている。プロセス管理用としてPCB (Process Control Block), エリア管理用として256ヶのACB (Area Control Block) を高速の内部レジスタ上に保持しており、プロセス切替えに伴なうこれらの制御レジスタの切替えを小さなオーバーヘッドで行っている。^(図4)Prolog実行時のメモリ管理の負荷が大きいため、

(ための)

高速化の種々の工夫がなされている。まず、スタック管理のための各種のスタック・ポインティングがABCで保持される。また、スタック領域は主記憶が使用されるが、スタック・トップの10教語は実質的に内部レジスタ上に置かれる。更に、実メモリの管理、GCC処理の各種機能はファームウェアで実現している。

4-13/4

3. フットウェア

PSIのオペレーティング・システム(OS)を*Sequential Inference Machine Programming and Operating System*と呼ぶ。ソフトウェア分野での数多くの進んだ考えを取り入れた、シングル・ユーザ、マルチ・プロセスのパーソナル(とあるが「プログラム内蔵環境」)OSであり、ウィンドウ・システムを始めとする優れた対話機能を提供する。SIMPOSはExtended Self-contained Prolog)と呼ぶシステム記述言語で記述されており、ユーザ・プログラムもESPで記述する。以下で、ESPについてまず説明し、次にSIMPOSの特徴を述べる。

3.1 ESP

ESPはPrologにオブジェクト指向に基づくプログラム・モジュール化能力を付加したものである。

Prologはユニティ・ション、バック・トラッキング、知識情報処理プログラムを記述する上で有効となる多くの長所を持つ言語である。しかし、もともとPrologは人工知能分野のみ規模のプログラム

作成を共同して設計されておる。

(効率よく記述し易く)

プログラムのモジュール化機能が無いほか、副作用を持つ処理の、そのままでは何等のシステム・プログラムの記述には適さない。この Prolog の短所と補なため、S-malltalk あるいは Lisp 上の Flacort 等のオブジェクト指向言語を参考にし、オブジェクト指向に基づくモジュール化機能を Prolog に組み入れた。

E-mP では、プログラムの枠組みをオブジェクト指向の機能で、プログラムのシステム系 Prolog スタイルで記述する事により、両者の利点を活かした強力な ^(オブジェクト指向) 強化を図る。

言語となっている。また、同時に、プロダクション・ルール・システムに基づく知識を記述できる知識表現言語としても効率的に利用できる。

E-mP によるプログラムの例を図 5-1 に示す。モジュールの単位はオブジェクト指向言語での呼称に従ってクラスと呼ばれる。^(オブジェクト指向言語では) 復作成のクラスの機能をそのま

ま、あるいは一部変更して取り入れたリ、あるいは機能を合成して取り入れたリである、継承と呼ばれるプログラム・モジュールの流用機能を提供しており、これによりプログラムの部品化と共有化ができる。E-mP もこの継承機能をサポートしており、S-mP のウィンドウ・システムのよう複雑なソフトウェアを効率よく記述しているほか、知識表現言語としても強力なものとなっている。

3.2 SEMPOS

SEMPOSのシステム構成を図6に示す。EMPによる単一言語システムであり、またオブジェクト指向にそった高度のモジュール化が施こされている。さらに、その強力な機能と比較して、簡明且つ拡張性に富み、階層化された内部構成となっている。

SEMPOSを構成するシステム定義のクラスもユーザ定義クラスもクラスとしては同質のものであり、オブジェクト指向に基づく統一されたアクセス手段（メッセージ・パッシング）で呼出して使用できる。ユーザ・プログラムからシステム定義クラスも利用することも可能であり、このアクセス手法を用いて SEMPOSはその全機能をユーザに解放している、いわゆるオープンなシステムとなっている。

SEMPOSの提供する機能のうち、特徴的なものについて以下に述べる。
 ① 入力管理システム
 ファイル・システムではUNIX流の木構造のファイル・ディレクトリを提示している。ディレクトリ管理機能を独立、拡張して、ファイルだけでなく、

主記憶に生成された任意のオブジェクトに命名してディレクトリに登録して管理できるとしている。ファイルは1つのオブジェクトとして実現しており、このオブジェクトに read/write 要求をメッセージとして送る事でデータの読み書きが行われる。ファイルやディレクトリの複写、削除、表示等の代表的なユーティリティ機能を~~利用~~でさるよう、ファイル・マネージャが用意されている。ユーザはマネージャが表示するメニュー・コマンドやファイル情報のウィンドウを見ながら、マウスやキーボードから指示する事で、簡便な操作で所望の機能の実行も指示できる。更に、パイプ機能も提供しており、複製々のユーティリティ・プログラムも組合せて複雑な処理も行わせる事ができる。

ウィンドウ・システムはビット・マップ・ディスプレイ上に表示されるウィンドウの管理を主に担当するシステムである。複製々のウィンドウを相互に重なり合せて表示する機能も提供する。ウィンドウはマン・マシン・インタフェースの要^{必要}であり、ユーザは応用プログラムの作成時に好みや用途に応じた種々

ウィンドウ・システムでは

のウィンドウも利用しだくなる。この要求に応じられるよう、ウィンドウの構成部品である、パネル、メニュー、フォント等について各種の仕様のものを取揃えている。ユーザはE&Pの多重継承機能も使用して、望む仕様の部品と取揃えて組合せる事により、意図するウィンドウも容易に作成できる。また、ウィンドウ・マネージャが用意されており、生成されてディスプレイ上に表現されているウィンドウに対して、位置やサイズを変更したり、ウィンドウの重なり状態を変更したりする事ができる。

(iii) プログラミング・システム (スクリプト・システム) は標準的エディタの1つとして広く世の中で使用されている EMACS

←(図7)

に準拠して作られており、ウィンドウ・システムの利点を活かすべく、種々の拡張も行っている。また、日本文入力も行えるよう、カナ漢字変換機能も内蔵させている。また、構文を認識してテキストの編集を行なえる構文エディタも別座提供しており、ユーザは構文の文法を与えてカスタマイズする事も可能となっている。

E&P用の言語処理系プログラムとして、コンパイラ、デバッガ、ライブラリが用意されている。コンパイラはクラス単価でソース・プログラムをコンパ

イルし、解説実行コードあるいは直接実行コードを生成する。前者はプログラムのデバッグ用コードであり、デバッグにはリインタプリタに実行される。デバッグはプログラムの実行を制御したり、メモリ上のデータの値の表示/変更を行なう機能を提供しており、マルチ・ウィンドウを利用した優れたエディタ・インクフェースを実現している。

ライブラリは作成したプログラムをクラス単位で管理する機能である。システム定義クラスもユーザ定義クラスもライブラリで一元的に管理され、クラスの登録、呼出し、削除も行なう。ライブラリは、いわば“プログラム・ベース”と管理するシステムと言える。インクリメンタル・コンパイル機能や EBP のクラスの継承の解決もライブラリにより実現されている。

3.3 P&I用ネットワーク・システム (S&I M ネット)

米国では、各大学の大学、研究所同での計算機資源の共用、メッセーヂやデータの交換等に利用され、学問の進歩に大きく貢献してきている。日本では昭和60年に通信回線の開放がなされ、今後国内でも全国ネットワークを構築する環境が整ったと云える。S&I M ネットは、国内における人工知能分野の研究のインフラストラクチャ整備の一環として、ICOT、内述メ-カ^{のほが}、大学や研究所に設置されたP&I M ネットワークで接続しようというものである。(図8) 同一地域内はLAN (S&I M LAN) で接続し、地域間はNATのDMZ網を介して接続する。

S&I M ネットに接続されたP&I Mのユーザは、P&I M上で提供される予定の

←図8、

電子メール、トーク、リモート・ファイル、リモート・プリント等の機能を使用して、

地域内外の他のPとIとの間でメッセージツाय
ルの交換を自在に行なえるようになる。

さういふネットは昭和61年中に新作を開始する予定であり、その後も杆器や
性能の強化が継続してなされる計画である。研究者間の意見や成果の交換の円
滑化に貢献し、オ5世代コンピュータの発展を促進する事が期待される。

(パソコンの研究)

4. おわりに

P&Eは世界で最初の本格的な Prolog マシンであり、そのハードウェア・アーキテクチャに多くの新しいアイデア、Pを盛り込むと共に、基本ソフトウェアにはソフトウェア分野での最先端の成果^(整然)を取り入れた野心的なマシンである。

実用に耐えうる性能、性能、信頼性も要求され、その開発は山あり谷ありの連続であった。しかし、ICCの強力な指導、同連メーカーの協力^(協力)、参加者全員の熱意により開発を成功に導く事ができた。

P&Eは昭和60年度末までに60台強が製造され、ICC、同連メーカー、のほかに、大学、公的研究所等に設置され、自然言語処理、エキスパート・システム等の各種知識情報処理分野での研究開発に利用されている。P&Eは今後も5世代コンピュータ・プロジェクトでのプログラム開発の中核マシンとして使用されていく予定であり、P&Eの性能の充実、高度化がなされるべく。また、P&Eを利用して行なわれる研究開発も広がりみせていくものとも予想される。

P512 を始めとするオラセ代コンベンター・プロジェクトの各種成果を広く
 世の中に普及し、知識情報処理の研究開発を促進して学術の進歩に貢献すること
 共に、実用化に向けて努力する事は本プロジェクトの目的である。P512 や
 P513 の P 512 も、知識情報処理分野の研究開発、実用化の基本ツールとして、
 国内外に広く普及させるための果敢たる挑戦も模索されつつあり、実現化する
 日も近いことと想われる。

(参考文献)

4) 内田俊一 他, 「コンピュータシステムを通じて見たインテレスの考察」, 情報科学 1985/8, pp25-31, サイエンス社

- 1) T. Chikayama: Unique features of ESP, Proc. of FGCS '84, pp. 292-298
- 2) K. Taki, et al: Hardware Design and Implementation of PSI, Proc. of FGCS '84, pp. 398-409
- 3) 内田俊一, 「知識情報処理システムの研究開発」, Computer Today 1985/7 No.8, pp. 10-17, サイエンス社
- 4) 横井俊夫 他, 「論理型言語と機械語として実行する逐次型推論マシンとその仕組み」, 日経エレクトロニクス 1984/11-5, pp. 173-200
- 5) 「特集: 知識情報処理」, コンピュータロール, 1985 No.10, コロナ社
- 6) 「特集: 知識工学」, 情報学会誌, 1985 No.12 Vol.26
- 7) Adele Goldberg: Smalltalk-80, 1984, Addison-Wesley
- 8) 米澤明憲, 「オブジェクト指向型プログラミングについて」, コンピュータ・ソフトウェア Vol.1 No.1, pp 29-41 ¹⁹⁸⁴

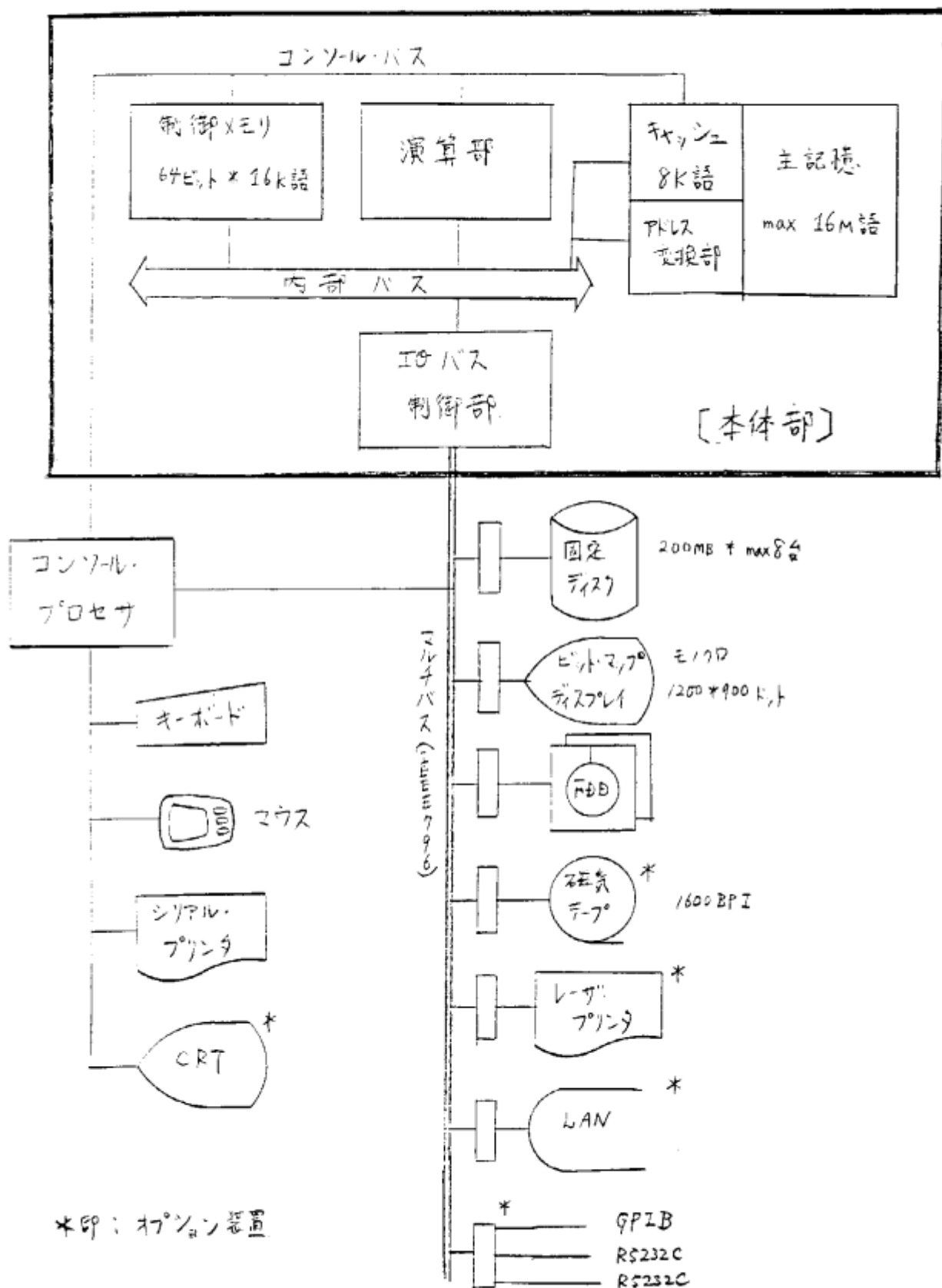


図1. PSIのシステム構成

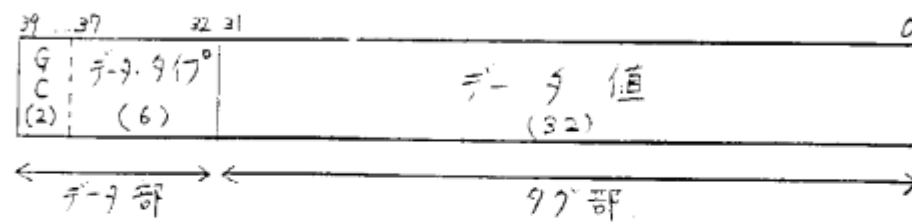


図 2 メモリ語の構成

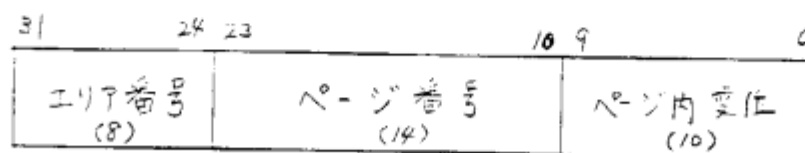


図 3 論アドレスの構成

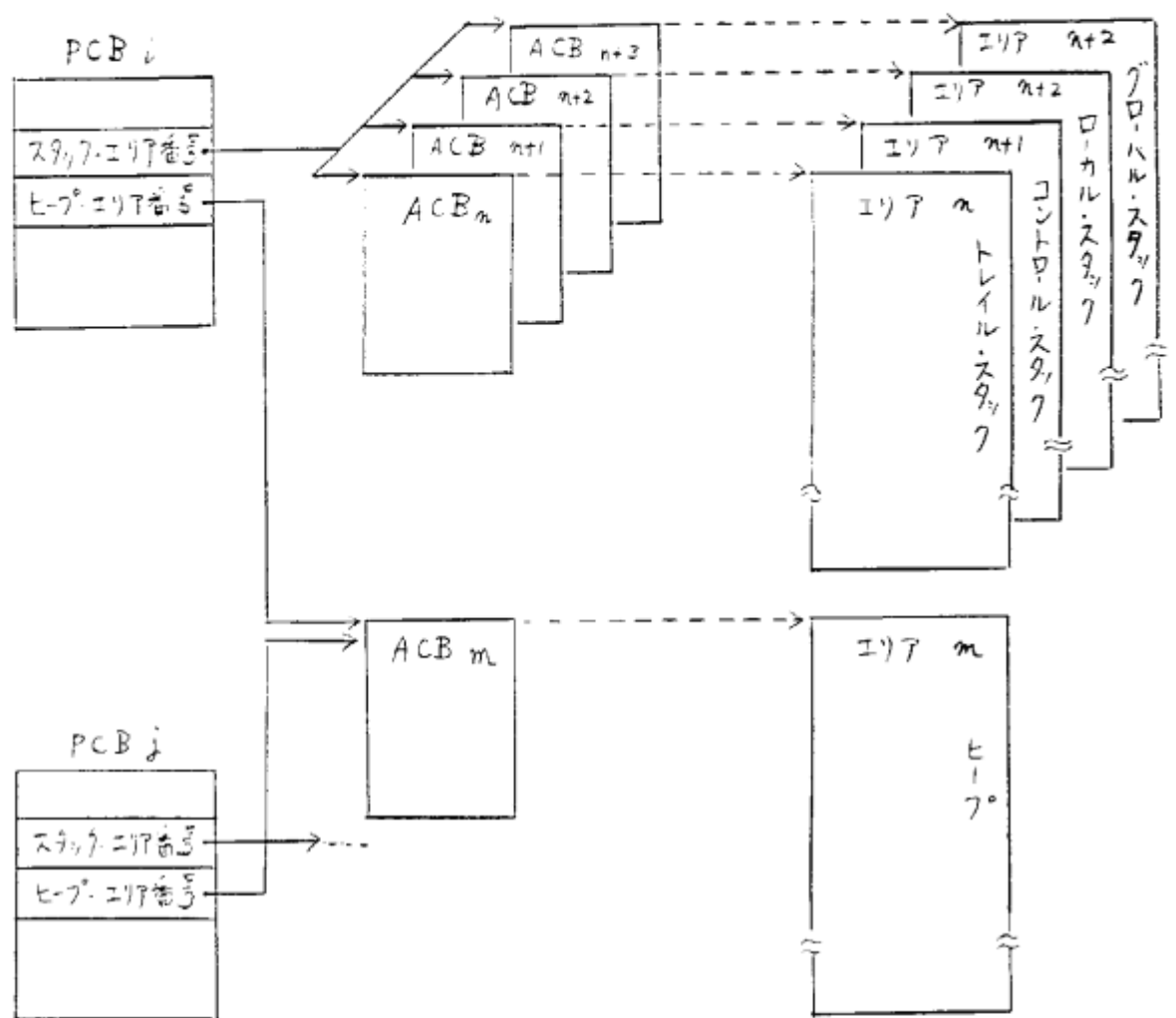


図4 プロセス管理とエリア管理の概要.

```

class my_window has      --- クラスの定義
  nature window:         --- 継承するクラスの記号

  :create_my_window(CLASS, Window) :-
    :create(CLASS, [size(900,250),
      position(100,100)], Window),
    :activate(Window);

instance
  component
    user_name,
    group_name;

  :message(Window, String0, String1) :-
    :write_lines(Window, String0),
    :write_lines(Window, String1),
    user_name(Window);
  :get_group_name(W, W!group_name);

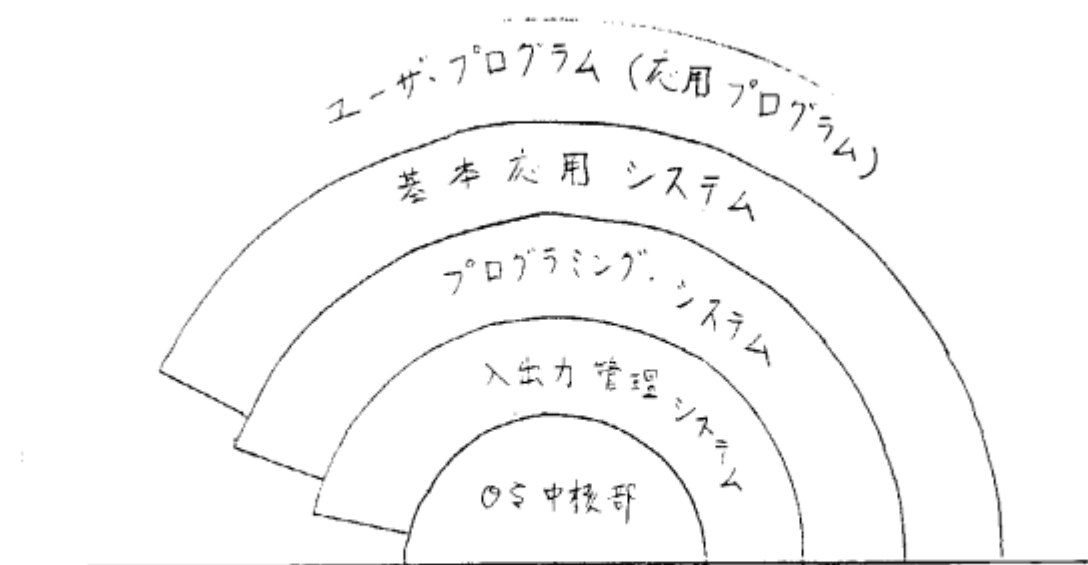
local
  user_name(Window) :-
    :write_lines(Window, "  : "),
    :write_lines(Window,
      Window!user_name),
    :write(Window, key#lf),
    :write(Window, key#cr);
end.

```

クラス定義の終了

クラス定義の終了

図5 EDP の プログラム例



- OS 中核部 : メモリ管理, 入出力装置管理, プロセス管理 他
- 入出力管理 : ファイル・システム, ウィンドウ・システム, ネットワーク・システム 他
- プログラミング・システム : エディタ, フォント・エディタ, コンパイラ, デバッガ, ライブラリ 他
- 基本応用システム : グラフィックス・システム, 文書処理 他

図 6 SEMPOS のシステム構成

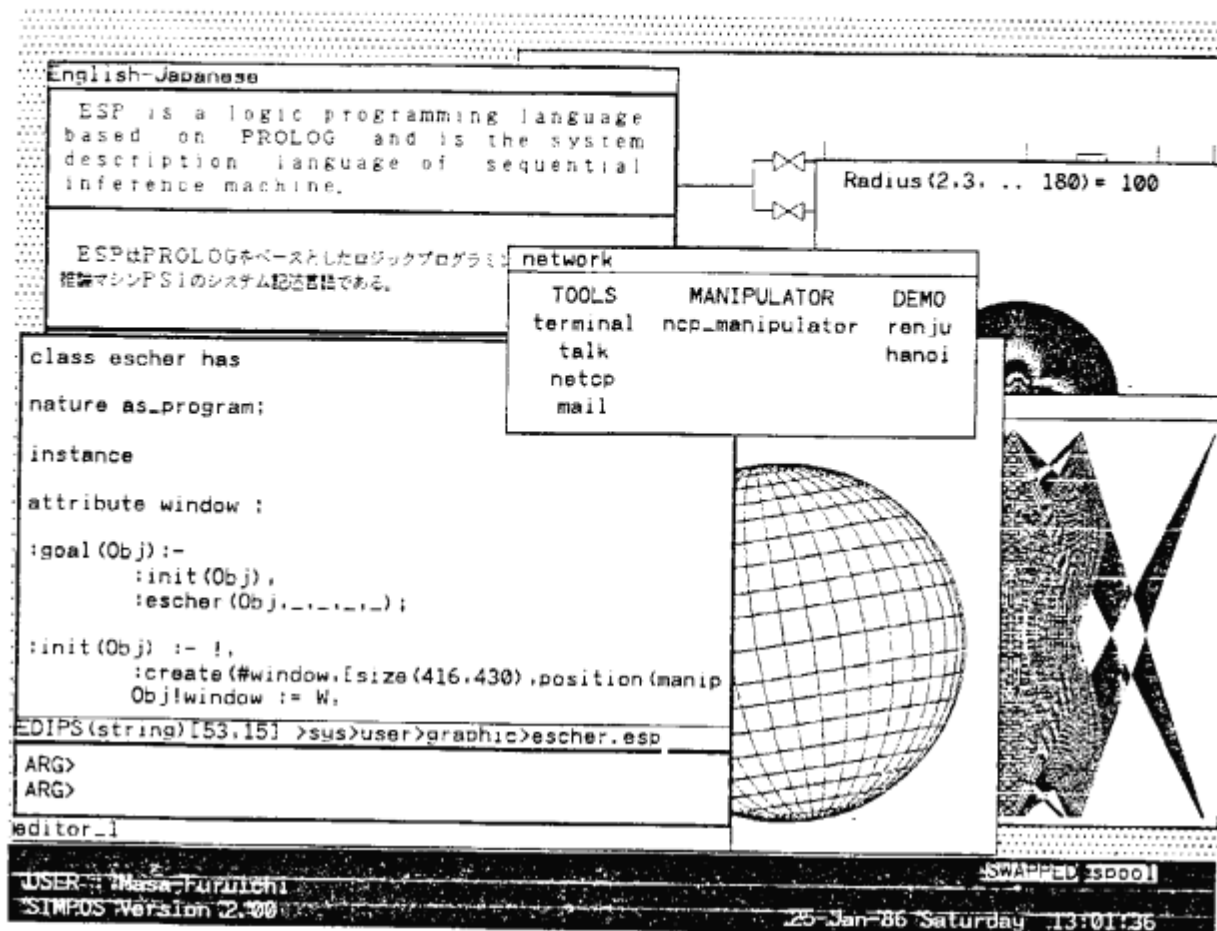


図 7 PS-E のウィンドウの表示例

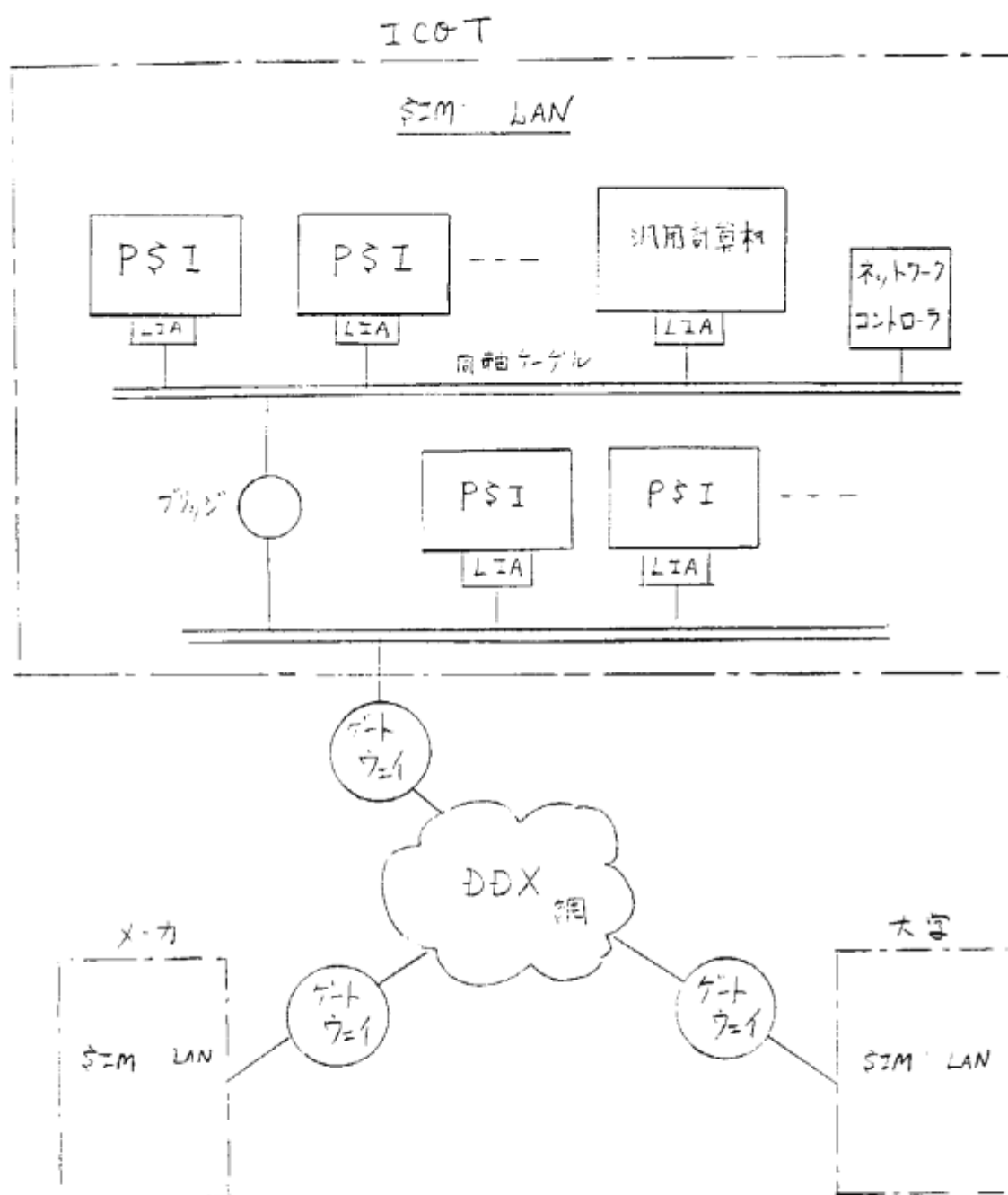


図8. SIM 網