

SIMPOS (逐次型推論マシンPSIのOS)
—ユーザ・インタフェースとその特徴—

佐藤裕幸, 近山隆, 石橋弘義, 吉田かおる, 内田俊一
(財)新世代コンピュータ技術開発機構

SIMPOSは、ICOTで開発された個人使用向逐次型推論マシンPSIのオペレーティング/プログラミング・システムであり、以下のような特徴を持っている。

- ・論理型言語にオブジェクト指向機能を付け加えたプログラミング言語ESPによりすべて記述されており、ESPの提供する継承の機能を用いることにより、ユーザはOSの機能を自由に活用することができる。
- ・ウィンドウ・システムを中心とした高度なマンマシン・インタフェースとESPに適した豊富なデバッグ機能を提供している。
- ・個人使用向のワークステーションの弱点を補うためにネットワーク機能を強化しており、資源の共有化を促進している。

SIMPOSは1983年5月から開発が開始され、現在、約40人のメンバーで第3.0版を開発中であるが、その規模はソース・プログラムの行数にして20万行を越えるものになっている。また、PSIはICOT、関連メーカ、大学等で利用されており、その数は今年度中に100台を越えようとしている。

本報告では、SIMPOSの提供する機能の概要及びユーザ・インタフェース、ネットワーク機能を中心に報告する。

SIMPOS (Sequential Inference Machine Programming and Operating System)
—User interface and its feature—

Hiroyuki SATO, Takashi CHIKAYAMA, Hiroyoshi ISHIBASHI,
Kaoru YOSHIDA, Syunichi UCHIDA

Institute for New Generation Computer Technology(ICOT)
4-28, Mita 1, Minato, Tokyo 108 JAPAN

SIMPOS is an operating and programming system of PSI(Personal Sequential Inference Machine) and it has following feature.

- ・SIMPOS is a unique language system based on a logic programming language : ESP.
- ・SIMPOS provides a good man-machine interface through a multi-window system and a useful programming environment for ESP.
- ・To make up a weak point of the personal work-station, SIMPOS provides powerful network facilities.

SIMPOS project was started in May 1983, the total number of source program is more than two hundred thousand lines.

This paper gives a brief overview of SIMPOS focusing on its user interface and network facilities.

1. はじめに

SIMPOS (Sequential Inference Machine Programming and Operating System) は、ICOTで開発された個人使用向逐次型推論マシンPSI (Personal Sequential Inference Machine, ψ) のオペレーティング/プログラミング・システムである。PSI はPrologの主要な機能 (ユニフィケーション、バックトラッキングとそれに伴う基本的なメモリ管理機能等) を機械語KLOの機能として備えている、高級言語専用計算機である。SIMPOSは、Prologにモジュール機能とマクロ機能を組込んだESPという論理型のプログラミング言語で記述されている。PSI/SIMPOS/ESPは論理型プログラミングのための良好な開発環境を提供し、現在、ICOT関連の多くの研究分野で利用されている。

本報告では、SIMPOSの提供する機能及びユーザ・インタフェースについて報告する。

2. SIMPOSの概要

(1) 特徴

PSI 及びSIMPOSは、論理型プログラミング言語による良好なプログラミング開発環境を提供することを目的として開発された。利用者 (アプリケーション・プログラム開発者) から見たSIMPOSの主な特徴は以下のような点である。

①開かれたオペレーティング・システム

全面的に論理型プログラミング言語ESP で記述されており、ESP の提供する継承の機能を用いることにより、ユーザはOSの機能を自由に活用することができる。

②高度なマンマシン・インタフェース

容易にカスタマイズできるマルチウィンドウ・システム等、高度なインタフェース機能を備えている。例えば、豊富なメニュー機能や編集機能を備えた (PMACS機能付き) ウィンドウ等、SIMPOSの実現に用いている部品をユーザに解放している。

③機能の豊富なプログラム開発環境

論理型/オブジェクト指向プログラム言語ESP によるプログラミングを容易にするため、強力なマクロ機能を備えたコンパイラ、プログラムの管理を容易にするためのライブラリアン、シンボリックなステップ実行ができるデバッグ、複雑なデータ構造を解析しやすいインスペクタ等の豊富なツール群を提供している。

④強力なネットワーク機能

個人向のワークステーションであることの弱点を補うためにネットワーク機能を用いて、ファイルサーバ、プリントサーバ、メールサーバ、PSI 同士や汎用機との間のファイル転送等、資源の共用を促進する機能を提供している。

(2) システム構成

SIMPOSは、単一ユーザ、複数プロセスを基本とするオペレーティング・システム部 (OS) と、その上で使いやすいユーザ・インタフェースを提供するプログラミング・システム部 (PS) からなる (図-2-1)。

①カーネル (資源管理プログラム)

カーネルは、ハードウェア資源 (プロセッサ、メモリ、入出力装置等) の管理及びシステムの起動を行う。

プロセッサ : マルチ・プロセス環境を実現する機能を提供する。

メモリ : メモリの割当てやアドレス空間の管理 (GC 等) を行う。

デバイス : 接続される各入出力装置の管理及び制御を行う。

カナ漢 : カナ漢字変換機能を提供する。キーボード・ハンドラに内蔵させることで、SIMPOSのどこからでも漢字を入力できるようになっている。

IPL : システム (SIMPOS) 本体のロードや初期化を行う。

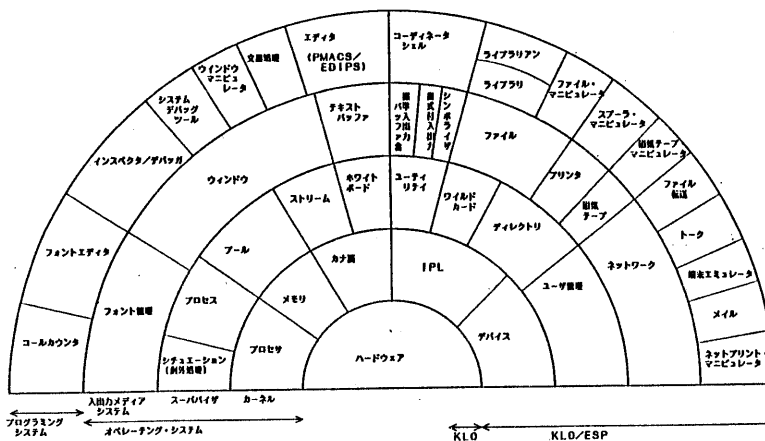


図-2-1 SIMPOSの

システム構成

②スーパーバイザ（実行管理プログラム）

スーパーバイザは、プログラムを実行するために必要な機能や環境を提供する。

プロセス：プログラムの実行主体であるプロセスの管理、制御を行う。

プール：オブジェクトを格納する種々の容器としての機能を提供する。

ストリーム：プロセス間の通信、同期の機能を提供する。

ワイルド

カード：ワイルドカード機能を提供する。

ディレク

トリ：ディレクトリの管理を行う。

ユーザ管理：ログイン・ユーザの管理を行う。

③入出力メディア・システム

入出力機能を提供する。

ウィンドウ：マウスやマルチ・ウィンドウ等の機能を提供する。

フォント：ウィンドウに表示するフォントの管理を行う。

シンボ

ライザ：プログラムやデータ等の外部表現（文字列）と内部表現（構造）との間の変換を行う。

ファイル：ディスク装置への格納、取り出しの機能を提供する。

磁気テープ：磁気テープ装置への格納、取り出しの機能を提供する。

プリンタ：プリンタ装置への出力機能を提供する。

ネット

ワーク：他PSIや汎用機とのデータ転送機能を提供する。

④プログラミング・システム

ユーザが直接操作するプログラム開発支援ツールとOSの各種ユーティリティの機能をメニュー選択という簡便な方法で提供するマネージャがある。各システムについては、4、5節で説明する。

3. ESP

SIMPOSはすべてESP(Extended Self-contained Prolog)で記述されている。ESPは、SIMPOSのシステム記述言語としてだけでなく、ユーザ言語としても広く使われている。

(1) ESPの機能

ESPは、以下のような機能を持っている。

①オブジェクト指向機能

オブジェクト指向という考え方を導入することにより、OSのような大規模なプログラムの開発には欠かせないモジュール化の機能を実現している。

②マクロ展開機能

ESPのプログラム中に現れる特定の文字列パターンに対してマクロ展開機能が働き、自動的に予め定められたパターンに展開される。これによりプログラム記述

を簡便にできると共に、プログラムの読み易さを向上させることができる。

③KLO言語機能

Prologの論理型プログラミング言語の他にベクタ等の拡張データ型、リモート・カット等の拡張制御機能、レジスタ・アクセス等のハードウェア操作機能を追加したKLO(Kernal Language version 0)の機能をすべて持っている。

(2) ESPのオブジェクト指向機能

オブジェクト指向とは、抽象データ型の考えを発展させたものであり、データとそれに対する操作や手続きを一体化させたもの（オブジェクト）を基本とした考え方である。以下にESPの主なオブジェクト指向機能を示す。

・クラスとインスタンス

ESPでのプログラミングは、クラスを定義することにより行われる。クラスとは、同じ性質を持ったオブジェクトをひとまとめにして定義したものであり、プログラムの一つのモジュール単位となる。このクラスから生成された個々の実体をインスタンス（オブジェクト）と呼ぶ。

・多重継承

クラスを定義する時に他のクラスの性質（定義内容）を自分自身に取り込むことができ、これを継承と呼ぶ。ESPでは、複数のクラスを継承すること（多重継承）ができる。この継承機能により、クラス間に共通する性質をできる限りまとめて記述し、異なる部分のみ個別に記述することが可能となっている。

・メソッド

ESPの述語には、クラスの外部から参照することのできるメソッドと、定義されたクラス内のみで使用されるローカル述語がある。メソッドはオブジェクトの外部インターフェースであり、そのオブジェクトの振舞い（機能）を定義したものである。それに対してローカル述語は内部的に使用されるだけで外には見せないものである。従って、メソッドは継承により取込まれるのに対して、ローカル述語は取込まれない。メソッドには、クラス全体に共通な機能を定義するクラス・メソッドと個々のインスタンスの機能を定義するインスタンス・メソッドがある。

・スロット

オブジェクトの動的に変化する状態を保持するための変数をスロットと呼ぶ。このスロットは、論理型言語の論理変数とは異なり代入によりその値を変化させ、バックトラックによって値は元に戻らない。

(3) SIMPOSでのオブジェクト指向プログラミングの例

SIMPOSでオブジェクト指向メカニズムを用いた例として標準入出力（standard-input, standard-output）がある。これは、文字、行、構造等の単位での入出力インターフェースをデバイスに依存しない形で共通に定義したものである。行や構造の入出力は、特に最適化を行わなければ文字単位の入出力を用いて行うことができる。従って、ファイル、ウィンドウ、プリンタ等の各デバイス毎に文字単位の出入

力を記述し、標準入出力クラスを継承するだけで行や構造単位の入出力も行えるようになる。また、これを使用するプログラムは、各単位毎にインタフェースが統一されているので、入出力デバイスが何であるかを気にせずに記述できる。つまり、メソッドを送るオブジェクトを動的に決めることにより、入出力を行うデバイスを決めることができる。SIMPOSでは、この概念を入出力だけでなく、あらゆる部分に適用させたシステムとなっている。

(4) オブジェクト指向による特徴

・長所

- ①モジュール化が進み、プログラムの共有、既存のプログラムの流用が容易になる。
- ②新たな処理に対して手続きの記述の"変更"ではなく、"新しく"追加するだけでよい。
- ③概念や思考の整理がしやすい。

・短所

- ①メソッド呼び出しに対してどのクラスのメソッドを実行するのが実行時に決まるので、メソッドはローカル述語に比べて実行時のオーバーヘッドが大きい。これは、ファームウェアによるサポートと、内部的な処理をできるだけローカル述語により定義することで対処している。
- ②親クラスで継承に係わる変更があると、子クラスもコンパイルし直さなければならない。これは、実行時のオーバーヘッドをなくすために、継承解析をコンパイル時に行っているためである。
- ③機能単位にモジュールが構成されていないので、あることを行いたい時にどのクラスを使えばよいのかが判りにくい。これは、マニュアル(説明書)の充実により対処するほかない。

4. ユーザ・インタフェース

以下にSIMPOSのユーザ・インタフェースについて説明する。

(1) ウィンドウ

ウィンドウは、SIMPOSのユーザ・インタフェースの中核をなすものである。ウィンドウとは、ビット・マップ・ディスプレイのスクリーン画面上に表示される論理的な端末である。SIMPOSでは、スクリーン上に複数のウィンドウを表示することができ、ユーザは各ウィンドウで別々の仕事を行うことができる。また、ウィンドウとマウス操作を相合せることにより、メニュー選択という簡便な操作手法を提供している。

SIMPOSのプログラミング・システムでは、このウィンドウやメニューの機能を利用して優れたユーザ・インタフェースを提供している。

(2) ユーザ・インタフェースの統一化

SIMPOSのプログラミング・システムでは、ユーザ・インタフェースを統一させるために、以下のような規則を設けている。

①文字(列)の入出力を行うウィンドウは、総てPmacs機能付ウィンドウを使用する。Pmacs機能付ウィンドウとは、文字(列)の入力中にPmacs(後述)の編集機能が使え標準入出力ウィンドウである。これにより、入力中の文字(列)の修正がすべてPmacsにより行われるので、これらのユーザ・インタフェースが統一される。

②メニューを2種類に分類して、それぞれのマウス入力の意味を統一する。この分類は、メニューに表示する項目により以下のように行なわれている。

(1) 操作を項目とするメニュー

一般的に操作は固定なので、固定長のメニューを用いる。

(2) 操作の対象を項目とするメニュー

操作の対象は可変なので、可変長のメニュー(スクロールできるメニュー等)を用いる。

(3) ログイン処理

SIMPOSでは、各ユーザ毎に異なる環境を設定できるようにするために、初期化ファイル等によってログイン時に以下のような設定を行うことができる。

①システム・メニュー項目の設定

システム・メニューはSIMPOSのコマンド・プロセッサの役割を果たすもので、何時でも呼出すことができ、プロセスの起動、終了、中断、再開等ができる。メニューには、EXPERTSとMEDIA SYSTEMSの欄があり、前者のプログラムは複数個並行して呼出すことができ、後者は1つしか呼出せない。ユーザはこのEXPERTSの欄にプログラムを追加してシステム・メニューから呼出すようにすることができる。また、使わないEXPERTSのプログラムをメニュー項目から削除することもできる。

SYSTEM MENU		
Experts		MEDIA SYSTEMS
Debugger	Escher	user process list
Pmacs	Oakaki	window manipulator
File	Queens	network
Librarian	Squares	shell
Font_editor	Bounce	whiteboard
Edips		kanakan initiate
Spooler		garbage collection
		login
		shut down

②ディレクトリの論理名の設定

各ディレクトリに論理名を設定することができる。また、ディレクトリを指定しなかった場合に使われるデフォルト・ディレクトリの設定もできる。

③初期化プログラムの設定

ログイン時に実行させるプログラムを設定することができる。

④PSの自動生成の設定

ログイン時にエディタやデバッガ等のPSを自動的に生成し、指定した位置にそれらのウィンドウを表示するように設定できる。

(4) プログラム開発手順

SIMPOSでのESPプログラムの開発(作成, 実行, デバッグ)は、以下の操作の繰り返しである。

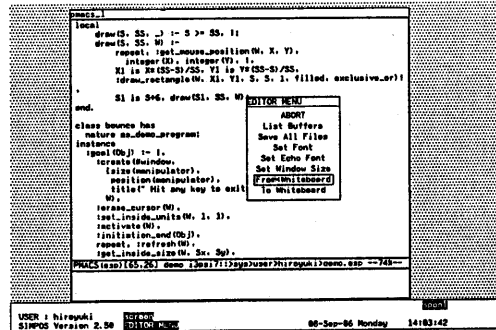
- ・エディタでソース・プログラムを作成, 編集する。
- ・ライブラリアンでクラスを登録し, 実行可能なコードに変換(コンパイル)する。
- ・デバッガでステップ実行し, デバッグする。また, オブジェクトの状態(スロット値)等のデータをインスペクタで調べる。

以下にこれらのPSについて説明する。

(5) エディタ

SIMPOSのエディタは、Pmacs と呼ばれる。Pmacs は、現在最も多くの人々に使用され非常に評判の高いemacs の特長を多く取り入れたスクリーン・エディタであり、以下のような特徴を持っている。

- ・100以上の豊富なコマンドを持っており、文字、単語、行、ページ等の単位で編集が行える。
- ・編集されるテキストを保持しているバッファは、どのプロセスからでも名前アクセスすることができる。従って、複数のPmacs で同じテキスト(バッファ)を編集できるし、逆に一つのPmacs で複数のテキストを編集できる。
- ・日本語入力機能(カナ漢字変換)が組込まれており、ローマ字をカナや漢字に変換できる。
- ・ユーザが好みに合わせてカスタマイズすることができ、また、ESPによりプログラミングすることによりコマンドを追加することができる。



(6) ライブラリアン

SIMPOSで扱うクラスは、総てライブラリが管理する。このライブラリの機能を提供するユーザ・インタフェースが、ライブラリアンである。ライブラリアンでは、以下のような機能を利用できる。

①クラスの登録, 削除

ESPのソース・プログラムから実行可能なコードを生成し、管理する。実行可能なコードには、解釈実行コードと直接実行コードがあり、どちらを生成するかを指定できる。また、登録済みのクラスを削除することもできる。

②実行コードの変更

実行コードとして、解釈実行コードと直接実行コードのいずれを使用するかを指定できる。

③テンプレート・ファイルのセーブ、ロード

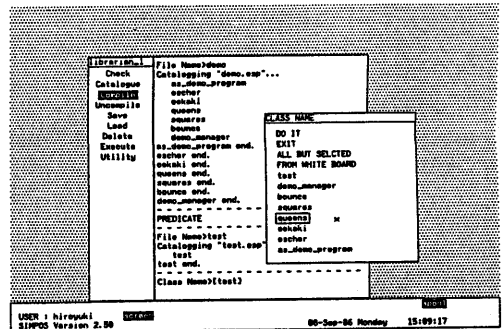
クラス単体の情報(スロット情報、メソッドやローカル述語のコード、継承クラス名、参照クラス名等)を持ったテンプレートをファイルにセーブしたり、主記憶にロードしたりできる。

④コマンドの一括実行

ライブラリアンへの一連のコマンドを一括して呼び出すことができる。これは、同じ組み合わせの多数のクラスを一括登録する時などに便利である。

⑤各種のユーティリティ機能

OSやユーザ定義のクラスに関する各種情報を検索することができる。



(7) デバッガ/インタプリタ

デバッガ/インタプリタは、ESPプログラムの実行及びシンボリックなトレース機能を提供し、解釈実行コード、直接実行コード、KLO 相込述語からなるゴール列を呼び出すことができる。

デバッガのトレース機能は、述語単位の操作に加えてクローズ単位の操作が行えるP&Cモデル(Procedure and Clause Box Control Flow Model, 図-4-1, 次ページ)により従来述語呼出しが失敗したということしか分らなかったのに対して、どのクローズで失敗したのかが分かるようになった。その他にデバッガは、あるゴール呼出しまでのトレースを行わないようにできるスパイ・ポイントの設定機能やマルチ・プロセス環境下でのデバッグ機能を提供している。

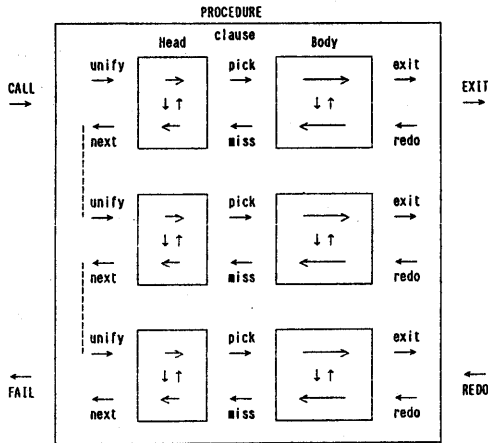
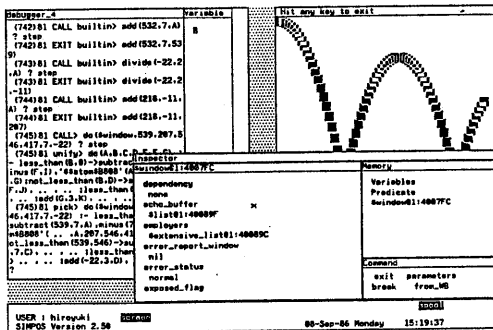


図-4-1 P&Cモデル

(8) インスペクタ

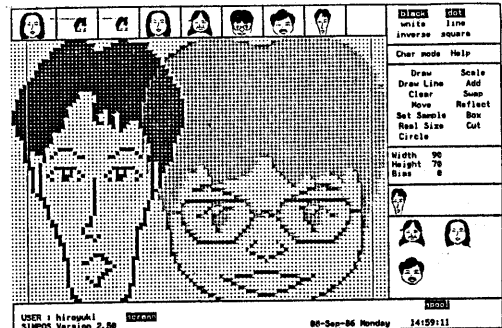
ESPプログラムのデバッグ方法は、デバッガによるトレースの他にオブジェクトの状態を確認する方法がある。この機能を提供するのがインスペクタであり、オブジェクトの状態(スロット値)やKL0の各種データ型の詳細情報の表示、検索、変更等が行える。インスペクタの表示はメニュー型式になっており、ほとんどマウスを操作するだけでデバッグを行うことができる。

インスペクタはデバッガから呼び出され、処理を終えるとデバッガに制御を戻す。このようにデバッガとインスペクタは密接な関係にあり、互いにデータの交換を行うことができる。例えば、デバッガから受取った変数の値をインスペクタで変更してデバッガに戻したり、あるいはデバッガの変数が値を持たない場合にインスペクタ側で変数に値を与え、デバッガに制御を戻して実行を継続させるといったことが可能である。



(9) フォント・エディタ

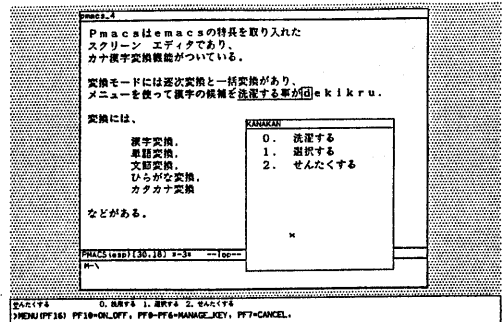
フォント・エディタは、ウィンドウに表示する種々の形状、サイズ、字種を持つフォントを対話的な操作で作成、修正するものである。フォント・エディタでは、システムが提供したフォントを変更したり、文字フォントを追加したり、まったく新しいフォントを作成したりできる。また、アイコンもフォント・エディタを使用して作成することができる。フォント・エディタでの操作は、総てマウスを用いて行うことができる。



(10) 日本語入力機能

SIMPOSで扱う文字コードは総てJIS 16ビットにより保持されており、英数字と漢字、カナを統一的に扱うことができる。SIMPOSでの日本語入力機能は、カナ漢字変換をキーボードに(ソフト的に)取り込むことにより提供されている。これにより、あらゆるプログラムでカナ漢字データをキーボードから直接入力することが可能となっている。

しかし、日本語の入力という観点からは、このような方法では使いにくいことも多い。そこで、SIMPOSのエディタであるPmacsにより、ワープロ的に日本語を入力できる機能が提供されている。Pmacsでの変換は、テキスト上の指定された範囲のローマ字を漢字やカナに変換することで行われる。漢字への変換の候補が複数個ある場合は、「次候



補]という形で徐々に表示することも、メニューにより一括して表示して選択することもできる。SIMPOSの文字入力を行うPSは、総てPmacs 機能付きウィンドウを使用しているので、Pmacs の機能で日本語を入力することができる。

また、Pmacs で入力した日本語文を指定されたレイアウトに乗っ取ってきれいに整形・出力する文書処理システムも用意されている。

(11) マニピュレータ

SIMPOSの各種ユーティリティ機能をメニュー選択という簡便な方法で提供し、以下のようなものが用意されている。

① ウィンドウ・マニピュレータ

ウィンドウの位置やサイズ等の各種属性の変更を行うことができる。

② ファイル・マニピュレータ

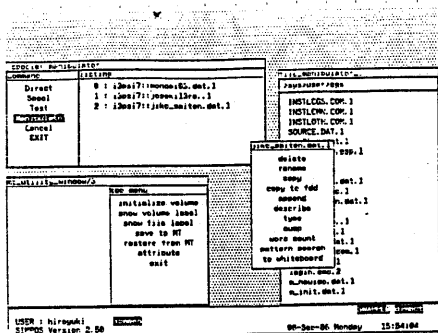
ファイル名の一覧表示、ファイルの複写や削除、パターン検索、ワード・カウント等の機能を提供する。また、ネットワークを介して、他PSI や汎用機のファイルにもアクセスすることができる。

③ スプーラ・マニピュレータ

スプールされたプリント出力の表示、取り消し、出力順の入れ替え等を行うことができる。

④ 磁気テープ・マニピュレータ

磁気テープの初期化、ディスク・ファイルの磁気テープへのセーブ/リストア等を行うことができる。



5. ネットワーク機能

SIMPOSのネットワークでは、LAN(Local Area Network) 及びNTT DDX 網を介しての広域のネットワークを提供している。同じ敷地内あるいは遠隔地のPSI や汎用計算機をネットワークで接続し、その間でメッセージやファイルの交換を行える(図-5-1)。

LAN はイーサネット(Ethernet)と類似した方式によるプロトコルを採用している。通信ケーブルとして同軸ケーブルを使用し、LIA(Line Interface Adapter)と呼ばれる接続装置を介してPSI をケーブルにつなぐ。ブリッジと呼ばれる装置を利用して、2個以上のLAN ケーブルを接続して論理的に1本のケーブルとすることもできる。

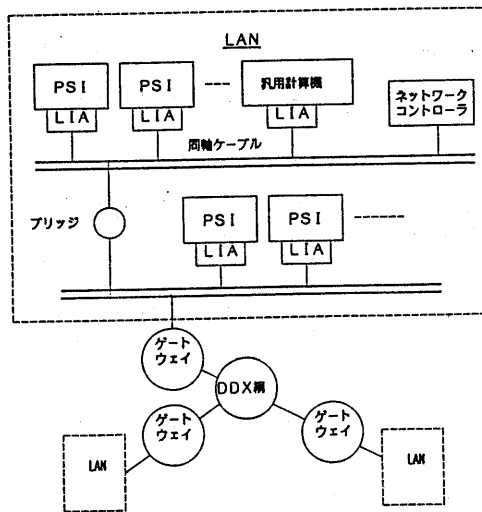


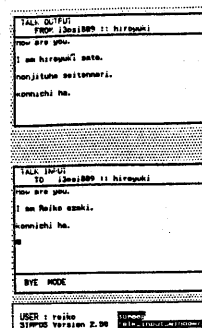
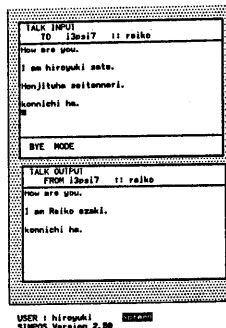
図-5-1 ネットワーク・システム

LAN をDDX 網につなぐ場合は、ゲートウェイと呼ばれる装置を介して行い、LAN とDDX 網との通信プロトコルの差異はここで吸収されている。PSI でのLIA を制御するプログラムをNCP(Network Control Program)と呼ぶ。NCP はより上位のアプリケーション層により高い論理レベルの通信機能を提供しており、この機能を利用して以下のようなシステムを実現している。

(1) トーク・システム

トーク・システムは、2台のPSI 間で会話を行う機能を提供する。PSI のユーザが他のPSI のユーザと会話(トーク)したい場合、トーク・システムを利用して相手側に会話を開始したい旨の初期メッセージを送る。相手側が会話を開始することに同意すると、以後トーク・システムを利用して会話を行える。

トークを開始すると、各PSI において入力ウィンドウと出力ウィンドウの2つが作られる。入力ウィンドウから入力したメッセージはそのまま相手側PSI の出力ウィンドウ



に表示され、同様に相手側PSIの入力ウィンドウに入力されたメッセージはそのままこちら側の出力ウィンドウに表示される。メッセージの送信、受信は双方向同時に(全二重で)行える。

(2) 電子メール・システム

電子メール・システムは、ネットワークに接続されている一群のPSI間でメッセージを手紙(メール)としてやりとりする機能を提供する。トーク・システムとは異なり、メッセージのやりとりは実時間ではなく、所定の受取箱(メール・ボックス)に手紙として配達される。SIMPOSの電子メール・システムでは、ネットワーク上のどれか1台のPSIにメール・ボックスを集中して置く方式を採用している。

メール送信に関して、以下のような機能が提供されている。

①送信メールの編集

メールを送信する時、Pmacsを呼出して編集できる。

②複数宛先指定

同一メールを複数のメール・ボックスに一括して配れる。

③受信メールの参照

各ユーザは自分の受取ったメールについて、メールの一覧表示、メール内容の表示、プリンタへの出力、ファイルへの保存・消去等を指示できる。

④メールの到着通知

ログイン時に新たに到着しているメールを通知する。また、PSI使用中に到着したメールは、その時点で操作に割込んでユーザに通知される。

⑤電子掲示板

複数のユーザが共同利用できるメール・ボックスが用意されている。このメール・ボックスに配達されたメッセージは、全ユーザからその内容を見ることが出来る。

(3) リモート・ファイル・システム

リモート・ファイル・システムは、ネットワークに結合されている他PSI上のファイルを、自PSIのプログラムから自PSI上のファイルとほとんど区別なく操作する機能を提供する。本機能を利用して、大容量のディスクが接続されているPSIをファイル・サーバとして利用することができる。また、他PSIや汎用機とのファイル転送も行える。以下にリモート・ファイル・システムの主な機能を示す。

- ①リモート・ファイルの作成、削除
- ②リモート・ファイルのオープン、クローズ
- ③リモート・ファイルの読み込み、書出し
- ④リモート・ファイルの状態問合せ
- ⑤他マシン間とのファイル転送(双方向可能)

(4) リモート・プリント・システム

リモート・プリント・システムは、ネットワークに結合されている他PSIのプリンタに、自PSIのファイルをプリント出力する機能を提供する。これにより、レーザ・プリンタのような高価なプリンタを持つPSIをプリント・サー

バとして使用することができる。

6. 開発の経緯と規模

PSI及びSIMPOSの開発プロジェクトはICOT研究所の発足当時から開始され、現在は、SIMPOS第3.0版に向けて約40人のメンバーが開発に従事している。

(1) 開発経緯

1982/6	ICOT研究所発足	…KLOの設計開始
83/5	SIMPOSグループ結成	…SIMPOSの設計開始
/12	PSI 1号機納入	
84/3	ファーム一応完成	…基本部分デバッグ開始
/11	FGCS'84 デモ版	…PS部分なし
/12	SIMPOS 0.7版	…バグ多し
85/2	0.8版	…何とか使える
/4	1.0版	…一応の完成版
/9	1.5版	…ユーザ・ インタフェース向上
86/3	2.0版	…信頼性、機能向上
/7	2.5版	ネットワーク強化
/12 (予定)	3.0版	ユーザ・ インタフェース統一化

(2) SIMPOSの規模

現在のSIMPOS(2.5版)の規模を以下に示す。

・ソース・プログラム	… 232,202行
・クラス総数	… 1,380個
・述語総数	… 18,601個

7. おわりに

PSIはICOT、関係メーカ、大学等を含めて、今年度中に100台を越えようとしており、SIMPOSは現在、第3.0版を開発中である。今後、より使いやすいシステムにしていくために、ユーザからの要求に答え、いっそうの改良、拡張が必要であると考えている。

<参考文献>

- 服部他：SIMPOSのオペレーティング・システム
情処学会第29回全国大会、p285-304、'84,9
- 黒川他：SIMPOSのプログラミング・システム
情処学会第30回全国大会、p295-316、'85,3
- 石橋他：SIMPOS-ユーザ・インタフェースと開発過程-
情処学会コンピュータシステム・シンポジウム
p107-113、'85,12
- 石橋他：SIMPOSにおけるプログラミング環境
昭和61年電気・情報関連学会連合大会
p5.69-5.72、'85,9
- 近山他：SIMPOSのプログラミング環境
情処学会第33回全国大会、
40-1-40-12、'85,10 (予定)