# (5) 逐次型推論マシン（SIM）のソフトウェアシステム

ICOT 第3研究室長　横井　俊夫

逐次型推論マシンのソフトウェアシステムでありますSIMPOS(Sequential Inference Machine Programming and Operating System)について報告いたします。

SIMPOSの概要，即ち，中心となります研究開発課題，基本となる設計方針，その位置づけ，そしてSIMPOS第一版のシステムの構造と機能，最後にまとめとして，SIMPOSの開発の経緯と今後の計画について，順を追って説明をします。

まず，SIMPOSの研究開発課題は，大きく次の2つであります。ロジック・プログラミング言語によるシステム・プログラムの作成とロジック・プログラミング言語のための高機能なプログラム作成環境の開発であります。すでに色々説明されてきましたように，我々は，狭い意味のロジック・プログラミングにとどまるつもりはありません。オブジェクト指向言語や関数型言語の持つ良い機能をロジック・プログラミングの上に融合する事によって，十分な汎用性を持った言語を開発しようと思っております。そして，この言語は，適切なハードウェアのサポートがあれば，システム・プログラムをも十分書く事が可能であります。もちろん我々の試みは，まだその途中でありまして，今後，様々の改善が必要と思っております。

ロジック・プログラミングの有用性を主張するにしても，もう小さなtoy programを書いて議論する時代は終りました。十分な大きさのプログラムを大量に作成し，様々の角度から総合的に評価できるようにしなければなりません。そのためには，高度なプログラム作成環境と十分な処理能力が必要であります。良いプログラム作成環境とそれを利用しての大量のアプリケーション・プログラムや様々のユーティリティ・プログラムの蓄積が達成されて，はじめて，プログラム言語はその価値を広く認められる事になります。この2つの研究開発目的にそった成果の一端を，今，FGCS'84会場ロビーのデモンストレーションでお見せしております。お見せしておりますシステムは，デモンストレーション用に急ごしらえをしたシステムではありません。本年度末に，SIMPOSの第一版，その規模およそ8万ラインから10万ラインになると思いますが，ほとんどのものがデバッグの最終段階に入ってのります。これらのプログラムの中で，デバッグが十分にされたものを，デモンストレーション用にふさわしい規模にまとめ上げたものが，今，お見せしているものであります。

次に，SIMPOSの基本設計方針をお話しします。基本設計方針は5つにまとめられます。第一に，一様なフレームワークに基づいたシステム設計ということであります。ロジック・プログラミングに基づいて，マシン・アーキテクチャ，プログラミング言語，オペレーティング・システム，プログラミング・システムを統一的に設計しております。従って，システム全体が整った構成になっております。システムを作る側からは作りやすく，使う側からは，システムのどこにでも手を入れる事ができ，各々の目的に合ったシステムに作り換える事が容易にできます。第二は，パーソナルなインタラクティブなシステムであるという事です。SIMPOSは，多くのスーパー・パーソナル・コンピュータと同様に，パーソナルな高い会話機能をもったシステムとして設計されています。第5世

代コンピュータが提供する知的機能をもったシステムは，ほとんどがインタラクティブに会話しながら，人間の知的作業を支援するシステムとなります。SIMPOSの対話機能は，これらのシステムの土台を与えてくれる事になります。第三は，データベース機能の活用であります。ロジック・プログラミングは関係データベースとよくなじみます。SIMPOSのファイル・システムも，近い将来，関係データベース機能を取り入れるように設計されています。またSIMPOSは，システム自身が，ロジック・プログラミングの持つ高機能なデータベースを活用するように設計されています。現在，この試みは十分とはいえません。今後の大きな課題であります。第四は，ウィンドゥ機能の活用です。これは，自由度の高いマルチ・ウィンドゥ機能の提供です。第五は，日本語処理機能であります。現在，コンピュータは英語文化の上に成り立っております。英語以外の言語を使用する人々にとっては，非常に不便なものであります。コンピュータは，全ての人が自国語で使用できるようにしなければなりません。日本人にとっては日本語であります。SIMPOSは，ソフトウェアの日本語化を目指して，基本となる日本語を使う機能を非常に充実させております。ソフトウェアは，本来，それぞれの国の国語を土台として作られるべきであります。お互いの間の変換や交換を助けるのが，機械翻訳の技術であります。

我々は，SIMPOSを設計するにあたり，Lisp マシン，Smalltalk−80，Unix等を大いに参考にしました。これらのシステムは，各々の開発者のすぐれたセンスに基づいて，十分な時間をかけて作られたものであります。我々も，十分な時間をかけ，SIMPOSの改良・拡張を続けていくつもりであります。

それでは，SIMPOSの構成と機能を簡単に説明していきます。

SIMPOSは，大きく3つの層から成っております。Programming System，Operating System，ＥＳＰの層であります。ＫＬ−Ｏが，ＰＳＩの機械語ですが，現在は，システム記述言語ＥＳＰと一体化されています。それでは，ＰＳＩに近い方から順に説明をいたします。

ＥＳＰはExtended Self−Contained Prolog の略で，SIMPOSは，全てこの言語で書かれております。詳細は先ほど内田の方から報告がありましたように，明日の水曜日のテクニカルセッションで報告されますので，詳細は，はぶきます。

ＥＳＰは，ロジック・プログラミングの上にオブジェクト指向や関数型言語の機能を融合する，非常に現実的な一つの解であります。ロジック・プログラミングにプラスして，クラスとインヘリタンスのメカニズム，状態をもつオブジェクト，マクロ展開に基づく関数型言語の表現機能等を持っております。

次に，Operating System を説明いたします。Operating System は 3 つの層から成っております。Kernel， Supervisor，I/O Media System の3つであります。

Kernelの構成要素は，Process Management，Memory Management，Device Management，IPL(Initial Program Loader) の 4 つでありまして，ＰＳＩハードウェアとスーパーバイザーの間のギャップを埋める役割を果たします。

Processor Managementは，マルチプロセスの環境を提供いたします。Memory Management は，メモリ・スペースの管理とガーベジ・コレクションをつかさどります。Device Management は，入出力機器の制御を行います。ＩＰＬはＳＩＭPOSのブートストラッピングと初期化の作業を行います。今までカットされてましたが……

スーパバイザーは5つのモジュールから成っております。プロセス，プール，ストリーム，ワールド，クロック，5つであります。スーパバイザーはプログラム実行のための環境を提供いたします。1つのバチャル・マシンを作り出すもの， 理想的なバーチャルマシンを作り出すものと思って下さ

い。このバーチャルマシンと，ＰＳＩハードウェアのギャップを埋めておりましたものが，Kernelであります。

プロセスは，プログラムを実行するアクティブなエージェントであります。プールは，オブジェクトの格納場所であります。ストリームは，オブジェクトの流れるパイプでありまして，プロセス間の通信や同期のために用いられます。ワールドは，オブジェクトへのパスネームの集まりでありまして，これによりまして，実行環境の大枠が決められます。クロックは，日時を示す時計，ストップウォッチ，アラームウォッチから成っております。これらの要素によりまして，１つのバーチャルマシンの環境が作られます。

Ｉ／Ｏメディアシステムは，現在3つのシステムから成っております。ファイル・サブシステム，ウィンドウ・サブシステム，ネットワーク・サブシステムの３つであります。これらは外部ワールドとのインタフェースをつかさどるシステムであります。

ウィンドウ・サブシステムの特徴は，高水準のマン・マシン・インタフェースをサポートする事であります。多重ウィンド，階層化されたウィンド，マウスによるメニュの選択，自由に作りかえのきくウィンドと，このような機能を提供してくれます。これらの機能を使いまして，それぞれの目的に合ったマン・マシン・インタフェースを作り上げる事が可能になります。

ファイル・サブシステムは，データやオブジェクトを長い期間，長期的に記録するシステムであります。レコードやデータの蓄積，それから，ダイレクト，シークエンシャル，そして，キーによる参照，オブジェクト，及ち，クラスのインスタンスの蓄積，記録，それから，オブジェクトやファイルへのパスネームによる検索などが，ファイル・システムの基本的な機能として提供されます。これらの機能を使いまして，近い将来，関係データベースマシンの機能等も，この上に融合する予定であります。

ネットワーク・サブシステム，このシステムは，他のマシンと交信するために，３種類のインタフェースを提供します。マシン間の通信，離れたマシンの間での通信機能，プロセス間の通信機能，それから，離れたマシン上にあるオブジェクトに対する操作を行えるようにする機能，この３つの機能を使いまして，柔軟なコミュニケーションができます。その上に，いわゆる，ネットワークオペレーティング・システムが，構成される事になります。

現在，プログラミング・システムは，５つのサブシステムから成っております。コーディネータ，エディタ／トランスデューサ，デバッガ／インタプリタ，ライブラリ／コンパイラ，エクセプション・ハンドリングとヘルプ・システム，５つであります。

コーディネータは，Ｅ・プロセス（プログラミング・システムは，Ｅ・プロセス，エキスパート・プロセスの集合として，構成されております）を管理し，ユーザとの柔軟なインタフェースを提供するのが役割であります。及ち，ウィンドを通じてユーザのコマンドをＥ・プロセスに送る事，Ｅ・プロセスをシステム・メニュによって，作ったり，消したり，起動したりする事，それから，Ｅ・プロセスへの特別のコマンドを解釈し，実行する事，ホワイトボードというものを設けておりますが，ホワイトボードというものを介して，Ｅ・プロセス間の通信を助ける事，これらがコーディネータの役目であります。これらによりまして，非常に効率の良いプログラミング・システムが構成されております。

次は，デバッガ／インタープリタであります。このＥ・プロセスは，プログラムの解釈実行とデバッグ情報の提供を行います。及ち，プロシージャ・クローズボックス・コントロールフロー・モデルにもとづくデバッグ情報の提供，それから，解釈実行されるコードと，コンパイルされたコー

ド間の呼び出しを可能にする事，それから，マルチプルウインドを介して，様々のデバッグ情報をユーザに提供する事，これらがデバッガ／インタープリタの基本的な機能であります。

現在，ＤＥＣ—10 Prolog 等は，述語をデバッグの単位とみなしまして，ボックス・コントロール・モデルというものを提案しております。我々も，この精神に見習いまして，しかし，もう少し細部のデバッグ情報がほしいというところで，クローズをデバッグの単位として，このプロシージャ・クローズボックス・コントロールフロー・モデルを用いて，より細かなデバッグ情報を提供する，細かなデバッグ情報がとれるようにしてあります。ＤＥＣ—10 Prolog の精神を受け継いだというところであります。

次に，エディタ／トランスデューサ，これは様々な構造をもったテキストに対するエディタ群であります。プログラム・エディタ，テキスト・エディタ，フォント・エディタ等であります。プログラム・エディタは，ＥＳＰプログラムを編集するのに非常に便利になっております。また，マイクロ定義によりまして，いろいろシステムの仕組みを作りかえる事ができるようになっております。また，コマンド類は非常に少数にしまして，ユーザにとって使いやすくしてあります。それから，エラーに対する十分な対策機能，リカバリー機能を持っています。トランスデューサはエディタのサブプログラムの１つでありますが，テキストの変換，特に，ＥＳＰプログラム，ＥＳＰテキストのパージングとプリティプリンティングをつかさどっていますが，一般的に，構造をもったテキストをパージングしたり，プリティ・プリンティングをしたりするものとして設計されております。

ライブラリ／コンパイラは，ＳＩＭ上の全てのクラスや述語の管理をいたします。クラスの登録，プログラム・ファイルのローディング，コンパイル，インヘリタンス解析に基づくオブジェクトの生成等を行います。コンパイラはライブラリのサブプログラムとして，適時，呼び出されて機能します。

次に，エクセプション・ハンドリングとヘルプ・システムです。特にオペレーティング・システムやプログラミング・システムの重要な機能は，ユーザのあやまちやシステムのエラーに対して，柔軟に対処をする機能をもつことであります。これは非常に重要な事であります。それで我々は，一様な枠組に基づきましてエラーや，ヘルプ機能を統一的に扱うシステムを作っております。まだどのようなエラーがでてくるかわからないので，制限のない例外登録の機能，それから非常に柔軟な例外処理の機能をもったエクセプション・ハンドリングとヘルプ・システムを設計しております。

SIMPOS自身を開発するためのソフトウェア開発用のツールについてお話しします。SIMPOS のような，かなり大規模なソフトウェアを，しかも短期間に開発するためには，ソフトウェア開発用の強力なツール類をまず整備しなければなりません。我々はそのために，ＥＳＰクロス・システム，これはＤＥＣ—10上で動きます。実行時サポートシステム，イニシャル・プログラム・ローダ，システム・トレーサ等を開発して，これらを用いましてソフトウェアの開発を行いました。特に，ＥＳＰクロス・システムについて，若干，詳しくお話しします。ＥＳＰクロス・システムは，ＥＳＰインタプリタ，ＥＳＰクロス・コンパイラ，ＫＬＯクロス・コンパイラ，ＫＬＯクロス・リンケージ・エディタ，ユーティリティ類から成り立っております。重要な事は，これらのプログラムがほとんど全てといっていいほどProlog でかかれているという事であります。及ち，ソフトウェアの開発用ツールとして，十分のパーフォーマンスを提供できるだけの機能を現在の Prolog はもつに至っております。

次に，開発の歴史を簡単にお話しします。1982年，秋に，SIMPOSの設計に着手しました。1982年の年度末に，機能仕様書が作成され，1983年７

月に，ＥＳＰのオブジェクト指向機能の設計に着手しました。1984年末に第一版のクラス仕様書，どのようなクラスがあるかということのクラス仕様書を完成させました。1984年４月より，ＰＳＩ上で，単一プロセス環境でのデバッグが可能になり，1984年６月より，多重プロセス環境でのPSI上でのデバッグが可能になりました。入出力メディア・システムの主要部は９月に利用可能となり，プログラミング・システムは，現在ＰＳＩ上でデバック中であります。第一版は，本年度末にリリースする予定であります。

フューチャー・プラン，今後のSIMPOSにかかわる展開についてお話します。LISPマシンやUNIXの例でも明らかなように，プログラミング・システムやオペレーティング・システムは，長い時間をかけて育てていかなければ良いものとはなりません。SIMPOSは，今年度末に第一版を完成させ，来年度からは，研究開発用ツールとして本格的な利用が開始されます。本プロジェクトのソフトウェアに関する研究開発は，ほとんどがＳＩＭ上で展開されます。この使用経験を通じて，SIMPOSやPSIの不十分な点を洗い出し，改良・拡張を続けていきます。そして，ある時点で改良・拡張点をきちんと整理し，New SIMPOS として，再整理，再検討を行おうと考えております。また，SIMPOSの第一版は，プログラミングシステムとしては，必要最小限のものが用意されるだけであります。これに関しましては，大幅な拡充が必要です。この拡充は，モジュールに関する知識ベースに基づく保守，再利用支援システムや，ＳＩＭ利用コンサルテーション・システムなど知的プログラミング・システムの研究開発と結びつけられて，展開されます。また，オペレーティング・システムに関しましては，ファイル・システムとネットワーク・システムに大幅な拡張が必要であります。これらの改良，拡張作業は，ＳＩＭをツールとして利用する研究開発の作業と一体となって進められます。

わずか２年程で，SIMPOSがこれ程までに開発できましたことは，我々の研究開発の方針，計画が正しかった事もさることながら，開発に従事した諸君らの高い能力と志気に負うところが大であります。ＩＣＯＴ，そして三菱電機，日本電気，沖電気，松下電器，シャープからの約40名の諸君達であります。この諸君達の努力に感謝して，私の報告を終ります。ありがとうございました。