

(3) 基礎ソフトウェアシステム

ICOT 第2研究室長 古川 康一

ICOT 第3研究室長 横井 俊夫

ICOT第2研究室長の古川でございます。基礎ソフトウェアの前期成果について報告したいと思います。

始めに、本プログラムにおける基礎ソフトウェアの位置付けを明らかにします。

ご存知の通り、第5世代コンピュータが狙っている目標は、知識情報処理に適した、超並列コンピュータを作ることです。実はこれは大変困難な課題でありまして、ハードウェア・ソフトウェア・アプリケーションの各レベルにおいて、解決されねばならない問題が山積みしています。アプリケーションレベルでは、物事を分散化し、協調して解くような、そのような新しい、問題解決方式が必要となります。それから、ソフトウェアレベルでは、記号処理自身を並列に行うそのアルゴリズム及び言語の開発が必要となります。それから、ハードウェアレベルでは、そうした記号処理を、超並列に実行するメカニズム、あるいは、ハードウェア自身が必要となるわけです。

この、大変困難な問題を解決するものとして我々が設定した作業仮説が、図1に示されているような、知識情報処理と超並列コンピュータアーキテクチャの間を、ロジックプログラミングを用いて橋渡しをするという事でございます。言い換えれば、その間のギャップを、ロジックプログラミングを用いて埋める、という事になるわけです。で、この仮説を検証する事が必要になるわけです。そのためには、この間のギャップを埋める事のできるソフトウェアを実際に、ロジックプログラミングを使って開発して、その可能性を実証しなければなりません。そのようにして開発されたソフトウェア自身が、基礎ソフトウェアシステムであるという事ができると思います。

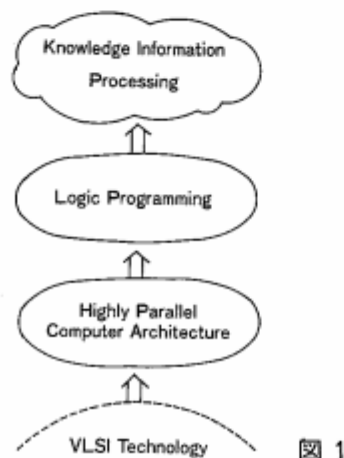


図 1

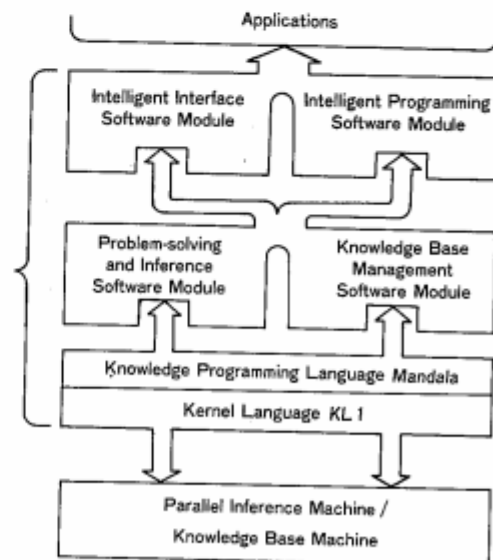


図 2

図2は、基礎ソフトウェアの構成を示したものです。この図で、一番下に、パラレルインファレンスマシン、ノリッジベースマシンがございまして、一番上に、アプリケーションがございまして、その間にはさまれた部分を我々は、基礎ソフトウェアシステムであると定義しております。これを下から順にみえますと、一番下にカーネルランゲッジというものがございます、その上に、ノリッジプログラミング・ランゲッジ、その上の上の形で、問題解決・推論を行うソフトウェアモジュール、知識ベース管理を行うソフトウェアモジュールがございまして、

以上の点について、私の方から報告いたしたいと思っております。それから、更に、その上に、インテリジェント・インタフェースと、インテリジェント・プログラミングのソフトウェアモジュールがございまして、それは後ほど横井の方から説明があります。

以下では、核言語、知識プログラミング言語、問題解決推論、知識ベース管理の各テーマ、及び応用システムについての成果報告を行います。最後に、今後の研究課題について簡単にふれたいと思っております。

最初に、核言語の報告をしたいと思っておりますが、本プロジェクトにおいて、核言語は大変重要な役割を担っております。それは、ハードウェアとソフトウェアのインタフェースをとるという事と、システム・インテグレーションの鍵になるという事でございます。我々は、核言語が、プロジェク

Kernel Languages

Initial Stage	Intermediate Stage	Final Stage
KL0	KL1	KL2
ESP	Mandala	

図 3

トの進行に伴って段々発展していくものというふうに考えまして、図3に示すようにKL0, KL1, KL2というシリーズを設定したわけです。で、KL0は、通常のプロログに近い言語になっております。そして、それは逐次型推論マシンSIMの機械語になっております。KL0の上に、ユーザ用のシステム記述言語ESPが開発されています。KL1は、並列型推論マシンPIMの抽象機械語と考えられ、大ざっぱに言えば、KL0に並列性を加えたようなものという事ができると思っております。で、KL1の上のユーザ言語として、知識プログラミング言語Mandalaを考えて設計を進めております。また、KL2ですが、これは、KL1に更に、知識ベース管理機能を強化したようなものというふうに想定しております。以下では、主に、KL1及びMandalaについて紹介していきたいと思っております。

KL1の特徴は、つぎの4つの点でございます。それは即ち、並列性、集合表現、それから、モジュール化及び、メタ推論機能でございます。以下順に、これら4つの機能について、説明したいと思います。

並列性ですが、並列性にはAND並列とOR並列の2つがございます。OR並列は、従来のPrologで、バックトラックによって解を全部探索する機能に相当します。この機能を用いますと、ゲームの探索のような多くの可能性の中から解を探し出すようなプログラムを記述するのに適しているわけです。一方の、AND並列性は、ANDゴール即ち、同時に満足しなければならない複数のゴールを並列に実行するという意味の並列性でございます。で、ANDゴールというのは、むやみに並列実行するという事はできないわけで、例えば、何人かで家を建てる時に、大工さんがはりきって、全部、最初に家を建ててしましまして、その後で、水道屋さんとか電気屋さんが工事しようと思っても、できないということになるわけでありまして、KL1で考えているAND並列性というのは、ストリームAND並列と呼ばれているものであります。

それは幾つかのANDゴールの間に、データの連続的な流れがあって、それによって、流れ作業的な並列が実現されることとなります。

一例をあげますと、私がいましゃべっているというプロセスと、通訳の方が通訳してくれて、それから、皆さんが私の話を聞くという一種の並列型問題解決を考えてみますと、私のデータをトランスレーターに渡し、トランスレーターの方が皆様にデータをサプライする、というそういうストリームになっているわけでございます。

この種の機能をもった論理型言語に、Concurrent Prolog, Relational Language, あるいは、Parlogといったようなものがございます。で、我々は、KL1を設計しておりますが、こういった種類の言語に基づいて、設計を進めております。

集合表現ですが、集合表現自身は、それをを用いて、関係あるいは述語を表現するのに用いられます。集合表現には、個々の要素を列挙する外延的な表現、それから、集合の各要素が満たすべき条件を記述する内包的な表現が許されています。で、集合表現の用途ですが、第1の用途は、先ほど述べました、OR並列部で得られる解の集合を表現するという事でございます。それから、第2の用途は、多少難しいのですが、述語自身が変数となるような高階論理の命題を記述する。これは、論理変数に集合表現を代入するようなそういった形のプログラムによって、実現されるわけです。

第3番目の特徴は、モジュール化を支援する機能でございますが、これはのちほど述べますMandalaのようなKL1上のユーザ言語にモジュール化機能を効率的に実現するという事を目的としております。モジュール化によって達成される機能には、情報の隠蔽でありますとか、分離コンパイル、プログラムの階層化、あるいは、一様でない世界を表現する多世界表現機能とかいったものがあります。で、こういった機能は、大規模なプログラムの開発を容易にする為には不可欠な機能と考え

られております。KL1で、この機能を実現するために、低レベルの機械語を一種の抽象データ型として導入しております。そのデータ化に付随する演算には、ソースコードからの機械語プログラムの作成、即ちコンパイル、コンパイルされたプログラムの呼び出し、或は、コンパイルされたプログラム間のリンケージをとるといったような演算がございます。

4番目の特徴は、メタ推論でございますが、この機能、今申し上げましたモジュール化機能化に基づいております。メタ推論機能自身には次のようなものがあります。即ち、ある指定された世界でゴールを解く、或はそのようにして解かれた結果に基づいて、判断を下す、或は、推論過程を制御する、世界を構成する知識を更新する、といったような機能でございます。で、これらの機能はすべて、計算の実行過程自身を扱う機能であると考えられ、それは、多重或は、階層化された世界を操作するメタレベルのプログラムとして実現されます。

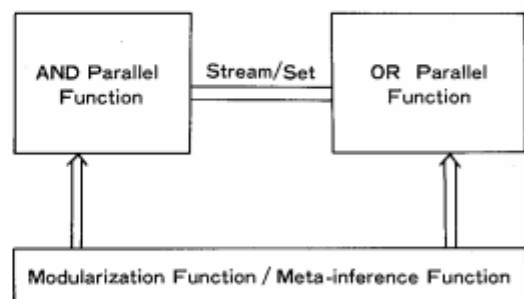


図 4

図4は、KL1の概念的な構成を示したものです。図の上の部分に、AND並列部とOR並列部がございますが、この2つは、集合/ストリーム変換によって結ばれております。即ち、OR並列部で得られましたある条件を満たす解の集合をデータのストリームに変換して、ストリームAND並列部に送って、ここで処理するわけです。

この間の通信自身は、別に一括して行う必要は

なく、ストリーム、即ち、たれ流し的に行う事ができるわけです。で、モジュール化、及び、メタ推論支援機能部分は上の2つの部分を共にサポートするというふうに考えております。

KL1を設定したことによりまして、超並列コンピュータアーキテクチャと、知識情報処理の間のギャップの、大体、半分を埋める事ができると思います。残りの半分のギャップを埋めなくてはいけない。で、それが以下に述べます Mandala の役割でございます。Mandala については、テクニカルセッションで発表いたしますので、ここでは簡単に述べてみたいと思いますが、第1の特徴は、図5に示す通り、プログラミング言語及び知識表現言語の2つが、単一の枠組みで実現されるということでございます。で、それは、論理

Mandala : A Knowledge Programming Language on KL1

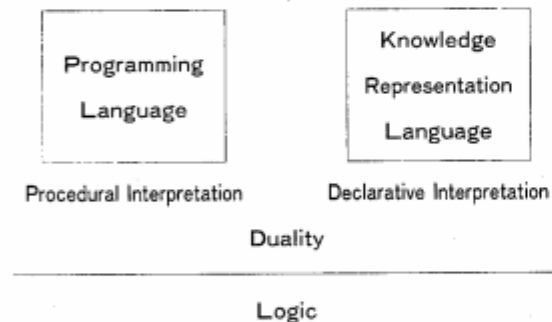


図 5

プログラミングの大きな特徴になっている「手続き的な読み方と、宣言的な読み方ができるということ」を直接反映しているわけです。即ち、手続き的な読み方に対応するのが、プログラミング言語としての側面、宣言的な読み方に対応するのが、知識表現言語としての側面であります。

Mandala の第2の特徴は、図6に示すようにプログラミング言語として見た時に、システムプログラミング言語、及び、知識プログラミング言語の2つを同じように、同一の枠組みで実現できるという事でございます。しかしながら、これらで

Mandala provides a single framework for both

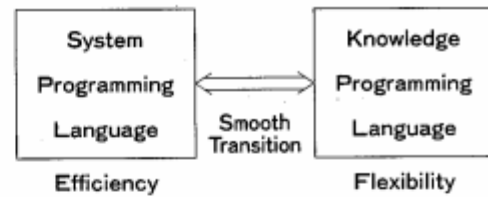


図 6

書かれたプログラムの扱いは異なるわけです。システムプログラミングにとっては、効率が大切であります。従って、その実行はコンパイルベースになります。一方、知識プログラミング言語にとっては、柔軟性が大切であり、従って、実行自身は、解釈実行ベースということになると思います。ここで大切な点は、その間、特に、知識プログラミングからシステムプログラミングへの移行が十分に円滑に行なわれる、という事でございます。そのためには、種々の最適化手法、例えば、プログラム変換ですとか、部分実行などの機能、或は、コンパイルを随時行う事ができるといった機能が必要となります。

Mandala の知識プログラミング言語としての特徴は、まず第1に、ロジックプログラミングを踏襲しているという事です。それから、最も大切な事柄は、KL1の高水準言語である、各種のプログラミングスタイルをサポートしているという事です。それは、オブジェクト指向プログラミング、或は、アクセス指向プログラミング（この2つは、ストリーム並列Prologに由来しています）、それから、ルール指向プログラミング（これ自身は、純粋Prologに由来している）、それから、メタプログラミングの機能がございます。メタプログラミングと申しますのは、自分自身のインタープリタを簡潔に表現できるという機能で、現在のProlog 或は、Concurrent Prolog などの言語に備わっている機能であります。で、その機能を用いますと、ユーザが簡単に自分でいろんな種類

の推論エンジンを定義できるということでございます。

知識表現としてみた時の、Mandalaですが、まず第1に、述語論理に基づいていることです。それから、プログラミング機能と同様に豊富な記述力を持っております。そこに書いてございますように、行動とか事象とかいったようなものの記述ですが、これは、ストリーム並列Prologのオブジェクト指向プログラミングに対応しております。それから、性質でありますとかルールを記述する能力は、純粋Prologがもっている、節単位でプログラムができる機能ということに対応しております。そういった特徴とともに、知識を管理するという事がございます。その中には、無矛盾性をチェックする、或は、冗長性を除去する、それから、実例を与えてルールを抽出する、知識を構造化するといった機能が含まれています。これらの機能は、具体的にはまだ実現されておられません、実現を計画しているわけでございます。

以上、お話ししました核言語及び知識プログラミング言語が、基礎ソフトウェアシステムの基本的な枠組みを与えるものと考えております。これからお話しいたします問題解決推論及び知識ベース管理は、その中で実現されるべき、より高度な機能となっていると考えております。問題解決推論機能で、我々がこれまでに焦点をあてて検討してきた主な機能は3点でございます。1つは、推論アルゴリズムの並列化、それから2番めは、問題解決における推論メカニズム、それから、より高度な推論機能を追求するという事でございます。

推論機能を並列化するという事は、その知識情報処理のアプリケーションから並列性をひきだすために非常に大切な技術であります。前期では我々は、ホーン節の演繹を並列実行するアルゴリズムの開発を行ったわけです。そこで開発されましたシステムは、OR Parallel Optimizing Prolog 略してPOPSと呼ばれております。POPS自身は、

Concurrent Prolog で書かれておまして、OR 並列計算が、ストリーム AND 並列計算によって実現されています。即ち、OR並列での可な解の集合を、ストリーム AND 並列でのデータ・ストリームとして表現する方法をとっております。POPSが開発された経緯についてですが、Prolog でチャートパーサーとよばれる並列型の構文解析機を開発したところ、それがほとんどOR 並列 Prolog インタプリタになっていたわけでございます。これは、sequential PROLOG 自身が、構文解析の研究から生まれたという事を考えますと、大変興味深いと思っております。POPSの実現アルゴリズムは、一種のグラフ・リダクションメカニズムになっておまして、重複計算を除去したり、または、サイクルをもったプログラムを扱えるという特徴を持っております。それから、OR 並列の複数解を生成する方式としてプログラムを無制限に走らせてしまうeager evaluationと、要求があってはじめて1つずつ計算を行う lazy evaluation の両方の方式を実現したわけです。

次に、推論の制御についてお話ししたいと思います。推論の実行をしていく時に大切な事は、問題をいくつかの部分問題に分割した時に、それをどのような順序で解いていけばよろしいかという問題でございます。それには、2つの方法がよく知られております。第1は、深さ優先探索と呼ばれておまして、1つの部分問題を深く追求していく、それが解にいたらない事がわかって、はじめて次の部分問題に進むといったやり方でございます。第2の方法は、幅優先探索とよばれておまして、各部分問題を少しずつ順繰りに解いていくという方法でございます。更に、推論の制御を行う方法には、この他、各部分問題ごとに、次に手がけるべき部分問題を指定してやるといった方法がございます。で、この指定方法自身をルール化してやる事もできるわけです。

このような機能を抽象化したのが、図7に示しましたdemo とよばれる述語でございます。この

(2) Meta-inference Function

○Control of inference processes

(∴ Algorithm = Logic + Control)

• demo (World, Goal, Control, Result)

○ Kaiser, LOOKS

• simulate (World, NewWorld, Goal, Control, Result)

○ KLI, Mandala

図 7

述語は、与えられた公理の集合Worldの中で、制御情報Controlを使ってGoalを解くと、その結果としてResultを得るという事を表わしております。本述語は、Bowen & Kowalski によって提案されたものでございます。この述語自身は Prolog によって、実現可能でございまして、そのプログラムは実は、Prolog でかかれた Prolog インタープリタとなっております。

後で述べます、知識獲得を支援するプログラムでございしますが、それはこのdemoに基づいて作られております。まったく同様の述語が Concurrent Prolog の上にも作られておまして、それは、Concurrent Prolog が、複数のオブジェクトの動作を記述する事から、simulate と呼ばれております。先ほど述べましたMandalaですが、その実験的処理系は、このsimulate に準拠して作られております。

問題解決推論の3番目のテーマでございまして、パワフル・インファレンス・エンジンですが、我々は定理証明機としてのロジックプログラミング言語の能力が限られておりますので、それを拡張するという事を考えております。それは特に、場合分けの推論ができないわけです。例えば、電話がかかからない場合に、相手の電話機のせい、或は、こちらの電話機が悪いかどうかですが、その時に、もし、他の所にかけてみてやはり通じないという事でしたら、自分の電話が悪いというふうに考える事ができると思っておりますが、そういった

問題解決の中で、それを解決するためのプランニングを行う能力というものを実現するためには、この場合分け推論の機能が必要になります。それは、1階述語論理の定理証明機によって実現されるわけです。

そこで、いくつかの証明方法について、Prolog を用いて開発してきたわけでございます。その1つは、線型導出、それから鍵導出、それから連結グラフ法といったものでございます。その他に、強力な推論エンジンの例といたしまして、項書き替えシステムの試作も検討しております。

次に、知識ベース管理の問題について、考えてみたいと思います。これ自身は、先ほど述べました問題解決推論機能とともに、知識情報処理を支える2本の柱の内の1本であると考えております。そこで考えてまいりました機能といたしましては、大規模知識ベースの実現方式の検討、論理学の枠組での知識獲得支援機能、知識表現システム的设计、およびエキスパート・システムの開発の4点でございまして。

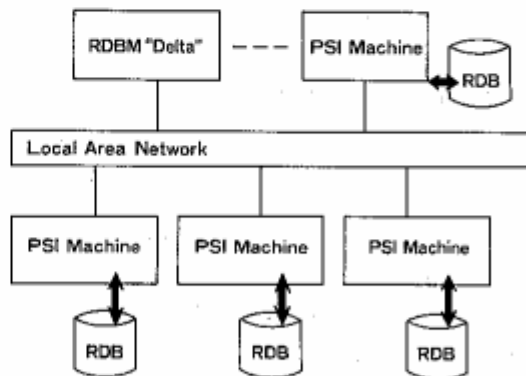


図 8

大規模知識ベースでございまして、その実現を行う第1歩として図8のようなシステムを実現するために、関係データベースマシンDeltaと、逐次型推論マシンPSIの結合ソフトウェアを開発しております。データベースマシンは、構造が単純であるが、大量のデータを高速に、検

索・分類する機能をもっております。一方、推論マシンは、小量ではあるが複雑な構造をもったルールを対象とした推論機能をもっております。これらの2つの部分系の各々の長所をうまく生かして、大規模知識ベースを実現するというのが本研究の目的でございます。そのためには、その2つの部分系をいかにうまく結合するかという事が問題になるわけですが、我々は、Prologで表現された質問文から、データベース・マシンへの問い合わせ命令列を生成するような新たな方式を考えております。それは、一言でいえば、Prologにおいて自分自身を呼び出す再帰的な質問を複数のコマンド列に変換するといった方法でございます。この大規模知識ベースの応用でございますが、それは、自然言語の辞書情報を格納することを考えております。

知識ベース管理の第2のテーマである、知識獲得についての主な課題は、知識ベースを無矛盾に保つためのいろんなサポートを行うという事でございます。で、その原理は、Prologをホーン節の証明機として使う、即ち、知識ベースを無矛盾に保つための条件である一貫性制約条件を、ホーン節で記述し、そのチェックに、Prologインタプリタ自身を使うという事でございます。知識獲得には2つのケースがございます。1つは、assimilationと呼ばれるものでございまして、入力の実偽が定かでない時に、それを一貫性制約条件でチェックしながらインプットを行うという事でございます。それからもう1つは、accommodationとよばれる機能で、入力が真である時に起るような矛盾を解消してやるという事でございます。で、このaccommodation自身は、Ehud Shapiro博士によるモデル推定システムを使う事を考えております。それから、これらの機能が、Prologの上で、メタ・プログラミング技術によって、統一的に実現されることが確かめられました。

次は、知識表現でございますが、知識表現は、大変難しい問題でございます。我々が行いました

のは、そのモデル化のために、人間の問題解決方式を将棋を例に取って分析したわけですが、そして、その結果として、知識アーキテクチャというモデルを得たわけでございます。それ自身は、分散型の問題解決機の構造をも与えているわけでございます。その構成要素は5つあり、通信媒体である場、それから、主として大局的判断のための状況を把握する認識型の知識、それから、局所的問題解決のための、パターン的な記憶型知識、それから、問題解決の手順を与える制御型の知識、それから、問題解決の対象を表現する対象モデル（例えば、頭に描く将棋盤のようなものに相当するわけですが）、の5つを抽出したわけですが、

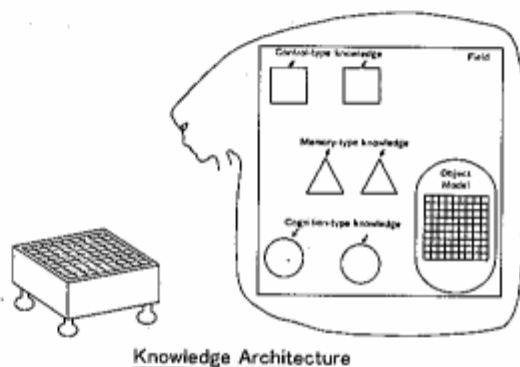


図 9

図9は、知識アーキテクチャの構成イメージを描いたものでございますが、この図自身は、人間が将棋盤に向かって考えている時を表わしたものでございます。頭の中にあるのが、知識アーキテクチャの構成イメージでありまして、場の中に、制御型知識・記憶型知識・認識型知識・対象モデルといったものが存在しているわけですが、

例えば、「将棋を指す」といった問題解決を考えますと、この時に、手を読む場合、手を読むのにまず必要となります順序制御をする制御型知識や、或は、手を生成する記憶型知識などが働くわけでございます。その際に、対象モデル内の局面は、手の読みに応じて更新されるということになります。

今まで述べました基礎的な研究と合わせまして、
 実用システムを開発するには、こういった機能
 あるいは、性能が必要となるのかという事を明ら
 かにするために、いくつかの例について、Expert
 システムを開発しております。その第1が、日本
 語の校正支援システムでございます。日本語の特
 殊性であります単語分け、或は、漢字、カナな
 どの様に、多くの種類の文字を適切に使用しな
 ければいけない。そういったものを計算機に支援さ
 せるのが、本システムの目的でございます。

具体的にどのような支援がどのように可能であ
 るかを調べるために、実際に、校正の専門家が行
 っている校正作業を分析したわけでございます。
 その結果といたしまして、ことばの誤りに関する
 訂正をするのに、使い方の基準や言い換え方の知
 識などを付加したような10万語程度の辞書が必要
 となるということが判明したわけです。それから、
 新聞記事などでは、固有名詞に関する情報がきわ
 めて有用であるという事も明らかになりました。
 それは例えば、人名に限っていいますと、20万人
 程度の人名・組織・ポストの組から成るデータが
 必要であります。これらの数十万件の比較的簡単
 な構造の知識を効果的に獲得、格納、検索するよ
 うな機能が必要となるわけでございます。

後の2つのシステムは、ともに、VLSI-CAD
 に関連したExpert システムでございますが、これ
 らはともに明日から始まるテクニカルセッション
 で発表がありますので、詳細には触れません。

図10に示しました、論理設計支援システムで
 ございますが、Occumという並列プログラミング言
 語でかかれた仕様から、DDLと呼ばれるハードウ
 ェア設計記述言語で表わされたオートマトンへ自
 動変換し、更にそれを、ロジック・ダイアグラム
 を経て、CMOSのセルレイアウトに変換するとい
 うシステムでございます。で、この各段階の変換
 にデザイナーの専門知識が使われている、とい
 うことです。特に最初の段階は、従来のCADステ
 ムになかったユニークなところでございます。

Logic Design Support System

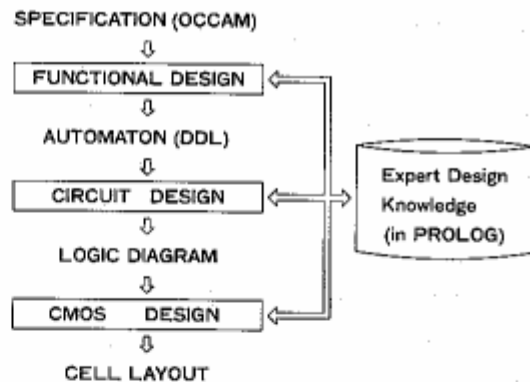
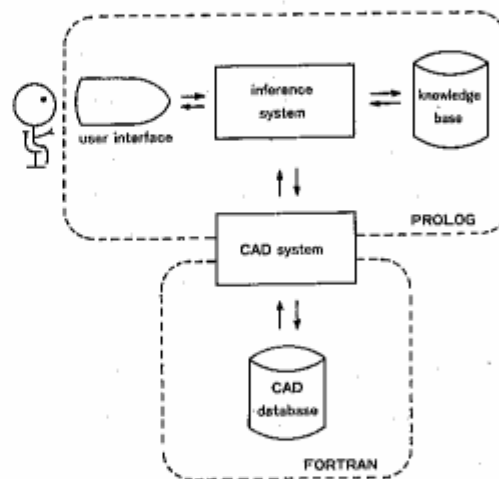


図10



Automatic Layout System

図11

図11は、VLSIの配線問題のエキスパート・シ
 ステムでございます。このシステム自身は、従来の
 CADシステムの高度化という事が主眼となってお
 ります。既存のCADシステムに組みこまれていま
 すアルゴリズム的な方法で、配線のうち98~99
 まで結べる、ところが残りが手作業として残さ
 れるわけです。その部分を、手作業を行うオペ
 ータの専門知識を組み込んだエキスパートシス
 テム

ムを作る事によりまして、半ば自動化するのが、このシステムでございます。そのため、本システムでは、PrologとFortranの結合を行っております。Fortran部分は、専用プロセッサと考えることができると思います。このようなシステムが、将来のアプリケーションシステムの1つのあり方になっていると思われまます。

最後に、今後の研究計画、すなわちこういった方向に研究を進めていくかについて、非常に簡単に述べてみたいと思います。まず核言語につきましては、SIM, Multi-SIM, PIM上での効率の良いインタプリタ・コンパイラを開発すること、それからKL1で実験的なソフトウェアの開発を行って、言語の表現力を評価すること、それから、AND並列性とOR並列性を一体化、緊密化することによってKL2の実現をめざすということを考えております。

問題解決・推論機能でございますが、1つは、並列問題解決機の開発、第2に協調型問題解決方式を追求する。これはともに並列性に関連しているわけです。それから、帰納とか類推とか非単調推論といったコモンスenseリーズニングに欠かせない、より高度な推論機能の追求をしていきたい、それから、これらの機能をMandalaに組込んで、並列実行環境における強力な知識表現システムを実現したい、こういうふうに思っております。

最後でございますが、知識ベース管理機能ですが、1つは、効率の良いMandalaの処理系を開発する、それから2番めにKAISERとよばれる大規模関係データベースシステムを分散データベース管理システムに拡張する、それから、KL1と関係データベースマシンの間のインタフェースを開発することを計画しております。さらに知識獲得機能もMandalaに組みこむことによって、並列実行環境における強力な知識表現システムの実現をめざしたいと思っております。簡単にまとめますと、前期は、基礎ソフトウェアの基本的枠組をデザインし、それから、その中に実現されるべきい

くつかの高度な機能の実現方法について、検討を行ったわけです。中期は、それらをより進めるとともに、1つのシステムにintegrateしたいというふうに考えております。

以上で私の発表を終らせて頂きたいと思っております。どうも有難うございました。

ICOT第3研究室の横井俊夫であります。基礎ソフトウェアシステムの知的マン・マシンインタフェースと、知的プログラミング・システムについて報告いたします。

まず、知的マン・マシンインタフェースであります。前期では、自然言語処理に関する研究のみが取組まれました。自然言語処理は、大変重要な研究課題であります。計画上、インターフェース部分に位置づけられておりますが、第5世代コンピュータの中心のメカニズムに深くかかれる重要な研究であります。本プロジェクトは、“言語”を中心に置くプロジェクトであると思えますが、その言語の中でも自然言語は重要な役割をにないます。現在日本語と英語を対象にして進めております。

現在の自然言語処理の研究活動は、大きく3つに分類されます。さまざまな構文解析システムの研究開発と、実験的な文脈理解システムの試作と電子化辞書（機械可読辞書）の開発であります。

構文解析システムの研究の目的は、高機能な文法記述言語と効率の良い解析機能をもつシステムを開発することにあります。我々のアプローチは、ロジック・プログラミングに基づいております。ロジック・プログラミング言語は、文脈自由文法に密接に関連しております。ホーン節は、文脈自由文法のルールの手続き的解釈を与えると見ることができます。このアプローチに基づいて、DCG (Definite Clause Grammar), LFG (Lexical Functional Grammar), GPSG (Generalized Phrase Structure Grammar) の3つを中心に、文法記述システムの研究開発を進めております。

DCGに関しましては、Prolog上にtopdownパーザがインプリメントされ広く使われております。が、我々は、日本語文の解析に適したbottom-upパーザを開発しました。BUPシステムは、電総研と東工大の協力を得て開発されたもので、bottom-up, depth-first のアルゴリズムに基づく効率の良い実用的な構文解析システムになっており

ます。このシステムでは、パーズングの過程の中間結果を記録として残すことにより最適化が行なわれております。このシステムは、BUP Translator, BUP Tracer, Epsilon-Rule Reducer, Cycle Checkerなど多くの文法開発用ツールから構成されております。現在、このシステムは、後で報告しますSIM上に移植されまして、SIM上に開発された最初の応用プログラムとなっております。

高度な文法記述のためには、より厳密に文法的関係を表現するための形式的システムに関する研究が必要であります。我々は、現在の言語学の新しい文法理論であるLFGとGPSGを候補として採用しております。LFGシステムは、LFGの計算機構として、F構造に関する等価性をPrologに導入することによってインプリメントされております。LFGの文法規則は、DCGの規則に支援され通常のDCGシステムとしてでも、BUPシステムとしてでも処理する事ができます。GPSGシステムを実現する方法は、現在、Working Group 3で、言語学者を含めて研究を行っております。GPSGの枠組みのための効率の良いパーズング・アルゴリズムの開発とGPSG上での基礎日本語文法の作成が中心の課題であります。以上の他にも、さまざまな考えに基づいた構文解析システムが試作研究されております。

次に文脈理解システムについて報告します。

文脈理解の計算モデルを確立することは、多くの難しい問題を含む重要な研究課題であります。本格的な文章理解システムのためにも、また柔軟で自然なインタフェースを実現するためにも、文脈理解に秩序立った取組みを行わねばなりません。会話のモデルには、少なくとも4つの重要な要素があります。speech act, coherency, presupposition, mutual belief であります。話し手と聞き手は、これらの要素を通信のための一貫性と最適化の原理に従って取扱っております。このことは、システムが強力な推論能力を備えていなければな

らないことを意味しています。

我々は次の立場に立って、文脈理解の研究を進めております。

第1に、会話の理解は、意志疎通のための一貫性と最適化が計算の原理となっている動的なプロセスであるということ、第2に、このプロセスの目標は、utteranceとmutual beliefのワールドモデルを両者の間で作り上げることであること、第3に、このプロセスは、ワールドに関する部分的な仮説を立て、話題や話し相手に関する知識に照し合せながら、検証する、このくり返しの過程であること、第4に、イベントやオブジェクトが色々な関係で結びつけられたものによって、現実のワールドは構成されているということ、第5に、作られつつあるワールド・モデルは、システムがもともと持っている知識にも影響を与えるということ、第6に、mental modelやactionに関する理論は、文脈理解研究にとって大切な役割をはたすということ、以上の方針に従って研究を進めるにあたり、大きな手掛として状況意味論を採用しました。この意味論は、自然言語の新しいモデル論的意味、特に文脈にわたる意味に対する新しいパラタイムを与えるものと注目されています。もちろんこの理論は完成されたものではなく、提案されたばかりの新しいもので、今後さまざまに改良、拡張、変化していくものと思われまます。しかし、文脈理解の研究をしっかりとした枠組の上に展開するための大きな手掛を与えてくれるものと考えております。

実際に試作しました実験的なシステムは、DUALSと名付けられており、来週ICOTのオープンハウスにて公開します。このシステムは、小学校3年生の国語の試験問題を題材として、文脈理解のメカニズムを実現するために何をどのように手掛ければよいかを正確に把握するために試作されました。従いまして、新しい構文解析システムや、新しい意味理論や問題解決の手法等の、各々の役割や相互の関係を実験し、具体的なデータを

得るためのシステムです。現在のところは、基本的な部分が出来上がったところで、まだまだ実験的なシステムです。今後、さまざまの実験が行えるよう改良、拡張を続けていく予定です。システムの基本部令の構成は、図1に示すとおりです。

EXPERIMENTAL DISCOURSE UNDERSTANDING SYSTEM

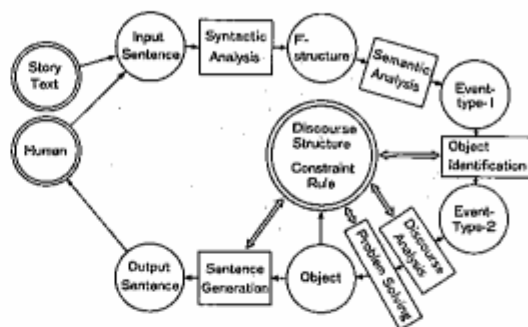


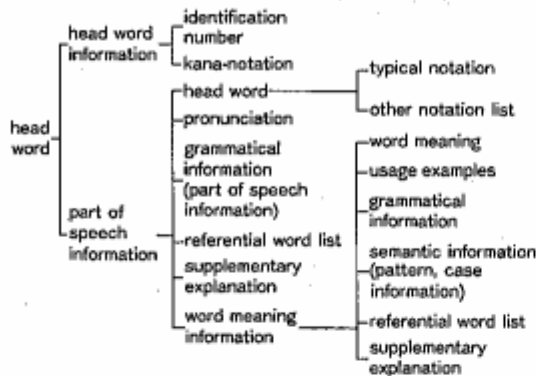
図 1

構文解析部はLFGを用いております。内部の意味表現は、状況意味論の基本的枠組に基づいたものです。その意味表現の仕方にそった簡単な文脈解析と問題解決の仕組みを採用しております。

次は、電子化辞書、機械可読辞書です。自然言語研究を効率良く進めるためにも、開発された自然言語処理技術を本格的な実用的なものとするためにも、十分な量の様々の言語データのデータベースの整備が重要であります。特に、電子化辞書の整備は急いで行わなくてはなりません。現存の、既存のテープ化された辞書は、そのままでは電子化辞書としては不適切であること、また著作権の問題があり、自由に加工して利用することができないこと等から、独自の基本的な辞書の開発に着手しております。本年度から3年の予定で日本語辞書、英和辞書、和英辞書、英語辞書の4種類のマスタ辞書を開発します。各辞書の規模は、約6万語前後で、各語彙につき、図2に示すような情報がつけられています。これは日本語辞書の場合ですが、他の辞書も同様になります。意味論的な情報や語用論的な情報は、現在研究を進めており

ますが、本格的に辞書に付加されるのは、マスタ辞書作成の作業が終了してからです。短期間に大きな辞書を開発するために、開発作業には、日本の代表的な出版社が加わり、既に集めている言語データを活用する形で進められております。

STRUCTURE OF JAPANESE DICTIONARY



知的インタフェースの中期の研究開発計画を簡単に紹介します。中期からは、自然言語処理に関するテーマは、より大規模に組織立って進められる事になります。また、音声理解と図形理解に関するテーマが新たにスタートします。自然言語処理に関しては、ドキュメントの様々な処理を行う技術やシステムの開発と、自然なマン・マシンインタフェースの開発が行われます。ドキュメントの処理に関しましては、翻訳をはじめとする、ドキュメントの変換、要約、検策、作成等の作業を自動化したり、支援したりするシステムを研究・開発することになります。そして、いずれに対しても、本格的な意味理解や文脈理解の機能を持たせることとなります。マン・マシシステムに関しましては、高機能な会話モデルの確立が中心になります。省略文の理解メカニズムと要点を的確にまとめた回答文の生成メカニズムの実現であります。この会話モデルは、次の音声理解の研究に結びつけられます。音声理解に関しましては、音響レベルの処理よりも自然言語処理との結びつけのところに重点をおいて進められます。図形理解に関しましては、ロー・レベルの処理よりも図形の知識

表現の仕方、それに基づく図形に関する推論、問題解決のメカニズムを明らかにするという方に重点がおかれます。線図形を対象に進めます。作図や製図の作業を自動化、支援するシステムを開発するとみることもできます。

以上の技術は、必要な部分を集め、各々の応用システムに適した統合的なマン・マシンインタフェースとしてもまとめられます。

次に、知的プログラミング・システムの研究開発について報告します。

現在の知的プログラミングの研究活動は大きく4つに分類されます。逐次型推論マシンの上に作成されますロジック・プログラミングに対する高度なプログラミング環境の開発、プログラムの仕様記述、検証、変換に関する研究開発、さらにもうひとつ、ワーキンググループ5を中心に進められていますCAPとACT¹の研究活動、そして、ソフトウェア・ハウス等ソフトウェア生産の現場が抱えている様々の問題に、知識情報処理技術を適用するための検討の4つであります。

まず、ロジック・プログラミングのためのプログラミング環境について説明します。この課題は、後のSIMPOSの説明の中で詳述しますので、ここでは概略のみをお話しします。この課題は、実用的な高機能プログラミング・システムを開発することと、将来の知的プログラミング機能のための具体的な土台を得ること、様々な実際のデータや経験を得ることを目的に進められています。

次に、プログラムの仕様記述、検証、変換について説明します。

本プロジェクトの大きな課題の1つが、まだまだ小規模な実験段階にあるプログラムの仕様記述、検証、変換の技術を実用的といえるところまで確立する事です。これらの技術を土台にして、はじめて、ソフトウェアの保守等の大問題に抜本的な解決を得る事ができます。現在、様々の方向からアプローチしています。仕様記述、検証、変換の3つは、互いに深くかかわり合っており、独

立ではありませんが、現在行われている研究活動をとりあえずこの3つに分類して説明します。

まず、プログラムの仕様記述に関しましては、記述言語のレベルとして、自然言語、形式的言語、プログラム言語自身の3つを考えて進めております。また、対象によりましては、図形や表を併用する事も有効であると考えています。自然言語による仕様記述につきましては、現在、東京工業大学のTELLプロジェクトの協力を得て、言語の設計、仕様記述を支援するシステムの開発等を進めております。

形式的言語による仕様記述に関しましては、1階述語論理や動的な行動の記述のための時間論理による記述などを実験している段階です。本格的な形式的仕様記述言語は、今後の課題であります。いずれにしても、自然言語、形式言語、プログラミング言語で記述されたものの相互間での検証や変換が容易になるためには、各言語の基本となる構造が同一である、或いは非常に類似していることが重要であると考えています。

プログラムの検証に関しましては、後で述べますCAPプロジェクトと、Prologプログラムに対する実験的な検証システムの試作が主な活動です。実験的な検証システムは、Boyer-Moore型のTheorem Proverから出発しまして、Prologの特性を生かした検証システムへと拡張されています。まず、Prologの実行そのものが一種の推論であるという事で、1階述語の推論は、Prologインタプリタの拡張形となっています。また、帰納法の適用は、関数型言語よりはるかに容易であります。Prologのために新しく開発しましたタイプ推論も有効に使われています。

プログラム変換に関しましては、まだ、非常に小規模な実験を行っているところですが、この技術は大変重要でありますので、今後はより本格的に取り組んでいく方針です。現在のところ、理論的な観点からは、等価性を保存する変換法の解明と、具体的には、Prologインタプリタ等いくつかの典

型例に対する実験を行っているところです。

次に特別のサブ・プロジェクトでありますCAP及びACT¹について説明します。ワーキング・グループ5は、第5世代コンピュータの基礎となる理論的な諸問題を解明するグループですが、現在、このグループを中心にしてCAP、ACT¹と呼ばれるサブ・プロジェクトが進められています。CAPはComputer Aided Proofの略で、数学の証明作業を支援するProof checkerを研究開発する事を目標にしています。研究を効果的に進めるために、またシステムを強力なものとするため、ある程度、証明の対象分野を限定する事が必要です。現在、線形代数、記号演算、総合微分幾何の3つを選択して進めています。線形代数は、取りかかりやすく、効果が明瞭に出るという事で選びました。現在、証明記述言語と証明システムの基本設計と試作を行っています。記号演算は、自己参照型の言語であるQuteの研究開発です。最後の総合微分幾何は、まだ新しい数学理論です。代数演算で総合幾何を行おうというもので、計算機上を実現するのに適した理論です。これは、まだ基礎的な理論調査の段階です。

CAPの最終目標は、定理証明や数式処理の技術を発展させ、実用的な証明支援システムを開発する事です。この成果は、プログラムの検証技術に類似のものとなりますので、知的プログラミングの研究テーマと深くかかわり合いながら進められることとなります。

ACT¹は、Theoretical Computer ArchitectureのT,C,Aを逆にしたというのが名前の由来です。プログラム言語、計算モデル、計算量、計算機アーキテクチャを総合的に検討し、第5世代コンピュータの計算機アーキテクチャや知的プログラミングの基礎を作るのが中心の課題です。現在、プログラミングスタイル、並列計算モデル、超並列計算機アーキテクチャに関して、体系立った整理を行っているところであります。

次に既存技術への転移について説明します。

実際のソフトウェア生産の現場と密なコンタクトを保つ事は、知的プログラミングの研究開発にとって大切な事です。現場が抱えている問題が何であるかを知り、また、開発された技術を現場に直ちに適用する、これらを通じて知的プログラミング技術は、実際に役に立つ、しっかりとしたもの成長していきます。もちろん、第5世代コンピュータの枠組は、ソフトウェアの生産現場が今従っている枠組とはかなりの違いがあります。第5世代コンピュータは将来のソフトウェア技術を目指すものです。しかし、様々のところですぐにでも効果の出る接点があります。例えば、ソフトウェア工学に知識工学を適用して高機能なツールや環境を作り出す事などです。現在は、このための特別なワーキンググループが作られ、様々な観点から検討を行っているところです。

以上、知的プログラミングに関する前期の研究開発活動の概略を説明しました。この活動は、現在は、様々な研究グループの活動の集まりのような形で進められています。ICOTの活動という面から見ると、この課題の重要さの割には、十分なパワーがさけなかったというのが現状です。SIMPOSの開発に力点を置かねばならなかったからであります。中期からは、重要課題として本格的な研究開発組織を作り、取組んでいく方針であります。

それでは、中期の研究開発を簡単にまとめて説明いたします。

まず、前期での各研究グループの活動は、より充実させ、一段と規模を大きくして展開されます。研究テーマとしては、やはり仕様記述、検証、変換の技術の追求が最大の焦点です。自然言語による仕様記述システムも、日本語、英語について、早く第1版を完成し、大規模なプログラムに適用し、実用性の検討が行えるようにしなければなりません。日本語の仕様と英語の仕様との間の機械的な翻訳も新しいテーマとなります。これらは、辞書類の整備もきちんと行なわなければなりません。これらの研究開発は、自然言語処理の研究と

密接に関連付けられて進められます。形式的仕様記述言語の設計も大きな課題であります。現在、いくつか候補を決め検討中ですが、早い時期に設計グループをスタートさせる予定です。プログラム言語としましては、後で説明しますESPをはじめとして、本プロジェクトの中で研究・開発する言語をすべて対象にして考えていくつもりであります。

検証と変換に関しましては、CAPの活動の成果と前期での実験システムの成果を統合し、より大規模な検証システムの試作に着手します。また、プログラム変換を重視し、より本格的なシステムの試作を行なうつもりです。

より具体的なアプローチとして、SIMPOSのプログラミング・システムの改良・拡張と新しい機能の研究開発も重要な課題であります。また、SIMPOSの開発に用いられたオブジェクト指向の機能を持つ論理型言語ESPの使用経験と実際にSIMPOSを構成するESPによるモジュール群も貴重な財産であります。これらを土台に、より知的なプログラム作成支援システムや、モジュール知識ベースシステムやSIMの利用や保守を支援するコンサルテーションシステムが開発されることとなります。

また、ある程度、プログラムの対象領域を限定する事により、高機能なシステムの実現を計ることも考えております。例えば、実時間システムを対象とする事がその一例です。

以上のような基本となる技術を開発し、最終的な目標は、Scott等の提唱するInferential Programmingの精神にそった、統合的なプログラミング・システムを開発し、ソフトウェアの開発、保守の問題に大きな解決を得ることです。

以上、基礎ソフトウェア・システムの知的インタフェース・システムと知的プログラミング・システムについて説明をいたしました。この2つの研究は、自然言語処理を介して深くかかわり合っています。また、これらの研究開発を中心にして、基礎ソフトウェアの研究開発は、次に紹介します

逐次型推論マシンSIMを有力なツールとして進められます。SIMにより、少なくとも中期の研究開発に関する限りは、十分な環境を得た事になります。もちろん、SIMはまだまだ不十分であります。基礎ソフトウェアの研究開発の過程で十分に使い込まれる事により、改良すべき問題点も明らかになり、拡張すべき機能もはっきりとしてきます。SIMの側から見ますと、これらの研究開発は、大量のアプリケーション・プログラムやユーティリティを用意してくれる作業でもあります。

以上が私の報告であります。有難うございました。