

AIシステム記述用言語 "COMMON ESP" の構想

2Y-5

中澤 修

山本 明

実近 憲昭

AI言語研究所

1. はじめに

AI言語として要求される最も大切な機能は、複雑な構造を持つ対象の知識を自然に表現できることである。そのために言語に対しては記号データの扱い、データの構造化、モジュール化、実行時に頻繁に生成、消滅が起こる大量のデータの領域の管理といったことが要求される。近年、このような観点から種々のAI言語が開発されると共に、その高機能化、実行の最適化、プログラミング環境の高度化等が盛んに行われている。

第5世代コンピュータプロジェクトにおいては、論理型言語を基礎とした種々のAI用言語の研究が進められている。その中で逐次型推論マシンPSIのシステム記述用言語として開発された Extended Self-contained Prolog(ESP)[1]は論理型とオブジェクト指向型を融合した言語であり、その記述能力等が評価され、我が国のProlog標準化案の基として提案されようとしている。しかし、ESPは現在のところPSI上でしか動作しない。我々が進めているプロジェクトはESPを基礎として、それをより高度化、汎用化した言語仕様を持つAIシステム記述用言語COMMON ESP (CESP)の開発とその仕様に基づき、一般の汎用計算機上で稼働する言語処理系の研究開発を行うことを目的としている。

本稿では、CESPの言語仕様、システム構成概要及び開発方針等について述べる。

2. 開発方針

我々のプロジェクトでは、以下に示す開発方針の下にCESPシステムの研究開発を行っている。

- 1) ESPを基礎とし、論理型言語とオブジェクト指向の特徴を有するESP upper compatibleな言語仕様の設定。
- 2) 各種汎用計算機上で動作可能なportability及び最適処理系の追及。
- 3) 他言語で記述された資源の利用。

各言語の特徴を生かしたプログラミングを可能

にするための使い易い、柔軟な他言語インタフェース。

4) プログラム開発環境の充実。

オブジェクト指向を利用した場合に、特に必要とされる統合的な開発環境及びユーザインタフェースの提供。

5) 効率的なメモリ管理方式、及び高速なガベージコレクション機能の提供。

6) オブジェクト指向機能におけるメッセージ通信の効果的な制御機構の提供。

7) 知識表現機能の向上及び並列パラダイムの導入。

3. 言語仕様概要

CESP言語仕様の基本となる概念はESPと同様に、論理型とオブジェクト指向パラダイムの融合である。すなわち、ホーン節の集合をオブジェクトとして定義することによって、手続きの構造化と階層化を行い、更に個々のオブジェクトに固有の内部状態を保持させるというものである。図1にその概念図を示す。

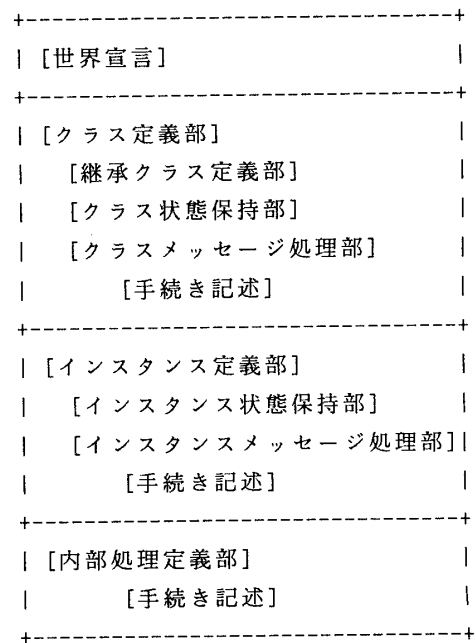


図1. CESP言語構造

The plan of AI system description language "COMMON ESP"

Osamu Nakazawa Akira Yamamoto Noriaki Sanechika

AI LANGUAGE RESEARCH INSTITUTE, LTD

CESPの基本構造は上記の形式であり、このフレームの中が論理型のプログラムによって構成される。

論理型の機能としては、以下の

* Prologの基本機能

* unification / cut 等に関する拡張機能

であり、オブジェクト指向の機能としては、論理型との整合性が問題であるが、「公理系 = オブジェクト」という基本方針の基に

* 多重継承機能

* メッセージ通信コンビネーション機能

を提供している。

他言語インタフェースに関しても、上記フレーム上でインタフェース用クラスとして定義し、他オブジェクトからのアクセスはすべてインタフェースオブジェクトに対するメッセージ通信として表現される。

その他当フレーム上で表現する機能として

* メッセージ通信制御機構

* 並列パラダイム

* 柔軟な記述能力を有する知識表現パラダイム

等を検討中である。

4. システム構成概要

我々が目標とするシステムの構成を図2に示す。

開発方針で述べたように、目標とする言語の性格上ユーザの利用環境を充実させることが重要であり、それが即プログラムの生産性向上につながると考えられる。そこでCESPシェルと呼ばれるサブOS的なものを作成し、統合的にシステム/ユーザプログラムの登録・抹消管理、実行管理を行う。

実行制御部においては、機種に依存しない処理系を想定しているため、仮想マシン語を設定し、それをエミュレートする方式を採用している。仮想マシン語としては現在WAM[2]をベースとして、それにオブジェクト指向実現のための拡張機能、各種最適化命令を導入している。

5. おわりに

以上、本稿ではCESPシステムの目標とするイメージについて述べた。

現在、実際の開発作業はPSIの処理系を基に、タグアーキテクチャを採用した専用マシン固有のデータ、メモリ処理を改良・拡張しつつ進めている。

また、プログラミング環境についてはCESPシステムをUNIX環境下で実現する方針で設計しているため、特にC言語とCESPとの融合方式に重点を置き開発を行っている。

今後は本稿内で述べた課題となっている項目の検討

[CESPシェル]

||

+--[システムローダ]

| |-シンボル管理

+--[プログラム管理システム]

| |-プログラム登録・削除

| | |-コンパイラ

| | |-リンカ

| | |-管理情報検索

| | |-世界(パッケージ)管理

| | |-継承解析ツール

+--[実行管理システム]

| |-デバッガ・インタプリタ

| | |-実行制御機構

| | |-例外処理機構

| | |-メモリ管理機構

| | |-インスペクタ

| | |-内部状態表示・変更ツール

| | |-性能評価ツール

+--[プログラムライブラリ]

| |-入出力管理

| |-データ構造管理

| |-プロセス・資源管理

| |-OSインタフェース

+--[システムジェネレータ]

| |-他言語コンパイル・リンク

+--[プログラム作成支援システム]

| |-オンラインドキュメント機構

| |-ヘルプ機構

| |-プログラム解析ツール

| |-プログラムCADツール

図2. CESPシステム構成

を行い、システムの試作版を完成させると共に、複数のプロセッサを並列的に利用する方向で進展すると考えられるアーキテクチャ技術の動向に従い、並列化処理可能な実行コードを生成するコンパイラの研究等も進めていく必要がある。

最後に、日頃御指導いただくICOT第4研究室近山隆氏に深謝する。

6. 参考文献

[1]T.Chikayama : ESP Reference Manual, ICOT Technical Report, TR-044, 1984

[2]D.H.D.Warren : An Abstract Prolog Instruction Set, TR309, SRI, 1983