

1 Q-5

エキスパートシステム構築支援ツール KBMS-3
- 構成方式 -服部 文夫 森原 一郎 木村 文宏
NTT 情報通信網研究所

1 はじめに

ES (エキスパートシステム) の開発がプロトタイプ開発のフェーズから実用ES開発のフェーズに移行するにつれて、故障監視システムと故障診断ESの統合など、既存情報システムとESの連携が重要な課題になっている。ES構築支援ツールKBMS-3では、このような背景から、従来Lisp言語処理系の上に作成していた開発支援機能や推論実行機能などを全てC言語ライブラリとして提供し、ESを既存情報処理システムへ組み込み可能にした。

KBMSのC化に当たっては、性能向上やメモリ使用量削減などC化のメリットを生かしつつ、動的な知識の変更やデバッグを可能にするなど開発効率を向上するための工夫を行っている。本稿では、KBMS-3におけるES構築支援機能の構成方式について述べる。

2 ES構築支援ツールに対する要求条件

(1) システム化の条件

既存システムへの組み込みを容易にするためには、単にプログラム間のリンケージを実現可能にするだけでなく、CPUやメモリなどのリソース制約をクリアする必要がある。そのため、処理速度の向上や実行モジュールのコンパクト化が必要である。

(2) 開発環境の条件

ESの開発では一般に知識の追加、修正を頻繁に行いながら知識ベースを作成することが多い。そのため、知識の追加、修正、デバッグのターンアラウンド時間を短縮することが開発効率向上のために重要である。

3 KBMS-3の機能構成方式

3.1 基本方針

C言語はLispなどのインタプリタ言語に比べ処理が高速でメモリ消費も少ない反面、コンパイルやリンクが必要なためデバッグ作業の効率が悪い。そのため、KBMSをC言語ベースにすることによってシステム化の条件を充たすことは容易

になるが、いかにしてシステム化の条件を損なうことなく開発環境の条件を充たすかが課題となる。そこで以下の方針でKBMS-3を開発することとした。

(1) 開発の効率化

手順の決まった処理についてはC言語で記述し、コンパイル、リンクを行うこととするが、修正の多い知識表現言語についてはインタプリタを提供し、部分的な修正/実行が即座にできるようにする。

(2) 実行(運用)環境のスリム化

単にインタプリタを提供したのでは実行モジュールが大きくなり、システム化の条件が損なわれる。そこで、開発支援機能と実行機能を分離し、実行環境で必要なモジュールを極力少なくする。つまり、知識のデバッグが終了した後は、実行機能のみを組み込んで利用可能な構成にし、実行環境のスリム化を図る。

(3) 機能のオプション化

KBMSでは、ルール、フレーム、表などの知識表現機能をはじめとして、ユーザインタフェース作成支援機能、分散処理支援機能、データベース利用支援機能などES構築に必要な多様な機能を提供している[1]。これらの機能を全て組み込むと実行モジュールが非常に大きくなり、システム化の条件が損なわれる。そこで、利用する機能に応じて必要なモジュールのみ組み込み可能にし、実行モジュールのコンパクト化を図る。

3.2 知識コンパイラと推論インタプリタ

実行の高速性と修正に対する柔軟性を共に実現する必要がある。そこで、以下の方式で知識コンパイラと推論インタプリタを実現した。

(1) ルールコンパイラは、複数のルールをマージして高速に推論が可能な実行形式(RETENETネットワーク)に変換する。このとき、全体を再コンパイルすることなしにルール単位の追加/削除を可能にすることによって、コンパイルの高速化を図っている。

(2) コンパイラによって作成した実行形式をそのままバイナリ形式で知識ベースに保存できるようにした[2]。これによって、知識デバッグが終了した後は推論インタプリ

タのみ組み込めばよいため、実行モジュールが大幅にコンパクト化できる。(図1参照)

(3) KBMSのルール知識表現では、記述の容易さと適用対象の拡大をねらいとして多様な表現機能を提供している。そこで、処理パターンによって推論インタプリタの部品化を行い、コンパイラはその部品を組み合わせる実行形式を作成する方式とした。これによって、利用する記述機能に応じて必要なインタプリタ部品のみ組み込むことができ、実行モジュールのコンパクト化が可能になる。

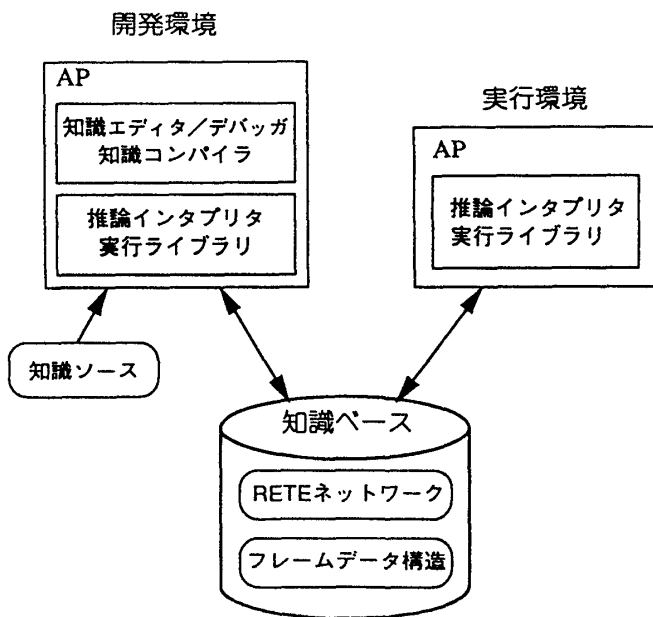


図1: 開発環境と実行環境の分離

3.3 手続きインタプリタ

すべての問題解決が純粋なルールだけで記述できるわけではない。ルールの条件やアクションにおいて、四則演算のようにどうしても手続きで記述する必要がある場合や、手続きで記述した方が簡単な場合がある。また、フレームにはメソッドやトリガといったフレームに固有な手続きを記述する。このような手続きはルールやフレームの変更に伴って変更される場合が多く、修正の度にプログラム全体のリンクを行っていたのではデバッグ効率が悪い。そこで、ルールやフレームに記述するための簡易手続き言語を作成し、インタプリタを提供した。

手続き言語には、四則演算のほか、whileなどの制御構造、ルールの実行制御やフレームの参照、更新などの知識処理用手続きが記述できる。また、利用者が作成したC関数を手続き言語として登録することも可能である。定型の手続きを予めC関数で作成しておき、それらを組み合わせる手続きを

記述することによって、手続き記述の簡易化と処理の高速化を図ることができる。尚、手続き言語は、ルールやフレームのコンパイル時に高速に実行可能な形式に変換している。

4 KBMS-3の機能構成

KBMS-3では、3章で述べた方式に基づいて開発支援機能および実行機能を全てライブラリとして提供している。ライブラリの構成は以下の通りである。(図2参照)

- (1) カーネル部: 知識ベース管理およびメモリ管理などの基本処理ライブラリ (必ず組み込む必要があり)
- (2) インタプリタ部: 推論インタプリタおよび手続きインタプリタ群
- (3) 知識コンパイラ部: 知識コンパイラおよび手続きコンパイラ群
- (4) オプション部: ユーザインタフェース作成用ツールキット、ESコネクタ (分散処理支援)、DBリンカ (データベース利用支援) など
- (5) 知識ベース作成支援部: 知識エディタ、デバッガ、手続き実行用シェルなど

上記構成によって、アプリケーションプログラムに必要な機能を適宜組み込んで利用可能になり、開発効率の向上と実行時のシステムリソース条件の充足が可能になる。

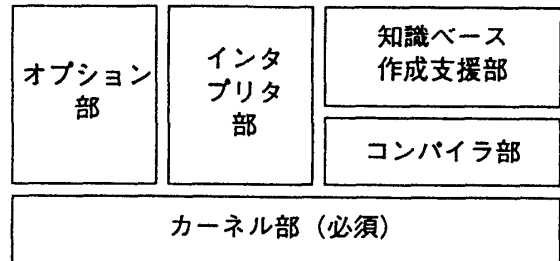


図2: KBMS-3のライブラリ構成

5 おわりに

本稿では、既存情報処理システムとの連携機能を強化したKBMS-3における機能構成方式について述べた。本方式によって、実行時の性能向上やメモリ使用量削減などのシステム化の条件と、開発効率の向上などの開発環境の条件を両立させることが可能になった。

参考文献

- [1] 服部他: KBMSにおけるES開発支援機能 - 概要 -, 情報 39 全大, 1989.
- [2] 村山他: エキスパートシステム構築支援ツール KBMS-3 - メモリ管理方式 -, 本予稿集, 1992.