

実時間問題向きシェル r t K D L の開発 (1)

4B-8

— システム構成 —

招行正 関口なぎさ 平松辰夫 高橋和男 水鳥哲也
(株式会社 明電舎)

1. はじめに

リアルタイム処理機能を有するエキスパートシステム構築支援ツール r t K D L について報告する。現在エキスパートシステムは本格的な実用段階を迎え、多様なエキスパートシステム構築支援ツールが開発されている。当社でもエキスパートシステム構築支援ツールである K D L を既に開発している[1]。しかし既存のシェルの多くや K D L は L I S P ベースで動作するか、あるいはリアルタイム処理機能を持たない。このため従来のツールでは、リアルタイム性(時間的に変化するデータの扱い、リアルタイム応答性等)を要求されるプラントの監視制御/運転支援への適用は困難である。

当社では、V A X シリーズの日本語 V M S 上で動作し、リアルタイムで高度な推論が行えるリアルタイムシェル r t K D L を開発した。本稿では r t K D L のシステム構成について述べる。r t K D L の知識表現と推論機構については別稿を参照のこと[2]。

2. リアルタイムシェルの必要性と問題点

監視制御の分野に従来の知識ベースシステムを適用するには次のような問題がある。

- 1) 非同期のイベント発生に対処できない
- 2) 条件やデータが動的に変化しない
- 3) G C によって推論が一時中断する
- 4) 外部環境とのインターフェイスがない
- 5) 推論速度が低く応答性が悪い

r t K D L では上述の問題に対して次の機能を満たし、特にデータの時間変化の知識を自然に表現できることやリアルタイム(実時間)応答性を重視している。

- 1) プロセス間通信(V M S システムのメールボックス、イベントフラグ)がルールで扱える
- 2) 時間変数、事象変数を表現できる
- 3) 独自のメモリ管理による G C の回避
- 4) 外部データの入出力機構を提供
- 5) R e t e アルゴリズムによる推論の高速化[3]

3. r t K D L の特徴

r t K D L の特徴として次の項目が挙げられる。

- ・ K D L の上位機種
- ・ フレームとプロダクションの融合
- ・ 高速な推論処理(R e t e アルゴリズム)
- ・ 時系列データの保存と統計量演算処理

rtKDL : A Real-Time Domain Shell = System Configuration =
Yukimasa MANEKI, Nagisa SEKIGUCHI, Tatsuo HIRAMATU
Kazuo TAKAHASHI, Tetsuya MIZUTORI
Meidensha Corporation

- ・ 事象変数
- ・ 推論時における実行要素の遅延実行
- ・ プロセス間通信(メールボックス、イベントフラグ)
- ・ ガーベジコレクションによる推論の中断がない

4. r t K D L の構成

r t K D L は図1に示すように開発環境の知識ベースエディタ、プリコンパイラと実行環境である知識ベース推論制御部、オブジェクト管理部、プロダクション制御部および内部時計、外部インターフェイスで構成される。

- (1) 知識ベースエディタ: 知識ベースの編集を行う。(文法チェックを含む)
- (2) プリコンパイラ: 知識ベースの C 言語への変換を行う。
- (3) 推論制御部: オブジェクト管理部とプロダクション制御部を統括するとともに外部データの取り込みや時間管理など推論全体の制御を行う。
- (4) オブジェクト管理部: フレーム機構の管理を行う。(メモリ管理、デバッグ環境を含む)
- (5) プロダクション制御部: プロダクションルールの推論制御を行う。(メモリ管理、デバッグ環境を含む)

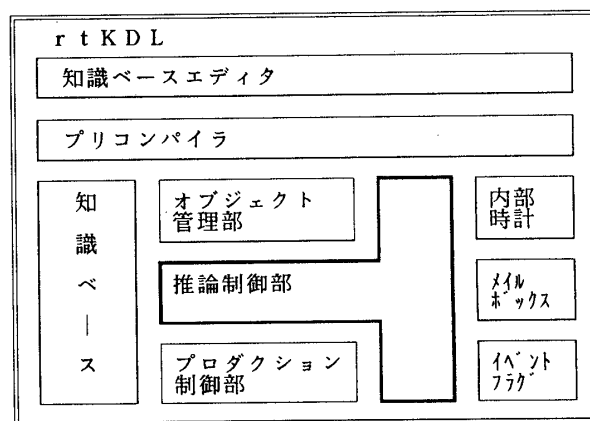


図1. r t K D L の構成

5. r t K D L の知識表現

r t K D L では K D L と同様にフレーム型知識表現と I F - T H E N 形式のルール型知識表現を統合し、目的に合わせた知識表現を可能にしている。

5.1 フレーム

フレームには上位概念を表現するクラスと具体的な対象を表現するインスタンスがある。クラスでは上位クラスと下位クラス間の階層関係に多重継承機能をサポートする。またスロットの特徴としてスロット自体の継承が2種類あり、定義体のフレームでしか参照できないもの

と下位フレームすべてで参照できるものがある。

rtKDLの特徴の一つである時間変数と事象変数はスロットを用いて表現される。時間変数ではスロット値が一定時間周期毎にスロットの持つ時間変数のバッファ領域にコピーされ、同時にバッファの統計量が計算される。事象変数では事象の開始・終了の状態と値が保存され、事象間の関係を表すことができる。

```

@class
(タンク :super          class
:document             "雨水タンク"
:class_slots
(最大容量             LONG :default 100000)
(最大排水量          INT  :default 150)
:instance_slots
(流入量              INT
:type                TIME_VARIABLE :buffer 360
:cycle               10000          :range 60
:compute             MAX MIN AVERAGE SUM)
(状態                STRPTR
:TYPE                EVENT_VARIABLE))

@instance
(
:class               タンク
:name                タンク-1
:slots
(最大容量           LONG :default 20000)
(最大排水量        INT  :default 50))

```

5.2 プロダクションルール

rtKDLでは複数の問題を並列に推論するための多重ワールドをサポートし、かつ一つのワールド内において複数のルールセット(複数のルール集合)を起動できる。ルールには目的に合わせた四種類のルール群、初期化ルール、終了ルール、メタルール、メインルールを持つ。ルールでは内部時間の参照、プロセス間通信のためのイベントフラグ、メールボックスの参照、更新、時間変数の統計量の参照、事象変数の事象の状態と値の参照、更新等が行える。またルールではC言語関数を呼出し、結果を評価することができる。

```

@ruleset( オーバフロー監視
:main
(rule1
(frame タンク/?tank
(:容積 ?vol :最大容量 ?max))
#[?vol > ?max]
-->
(fire (:context
(オーバフロー処置 :state ACTIVE))))))

```

6. rtKDLシステム

6.1 システム構成

rtKDLでは実行時における環境(実行システム)と開発時の環境(開発支援システム)を分離した。その理由として、1)実行時に開発環境を持たないことによる実行システムのコンパクト化、2)C言語化したことによる実行システムの高速度化、3)LISPで記述された高機能な知識ベースエディタとプリコンパイラによるエキスパートシステムの開発期間の短縮化が図れる。

(1)開発支援システム

rtKDLの知識ベースの作成を支援する知識ベースエディタと知識ベースをC言語に翻訳するプリコンパイ

ラの部分で構成される。プリコンパイラと知識ベースエディタでは文法のチェックが行われる。

(2)実行システム

プリコンパイラが生成した知識ベースのオブジェクトとrtKDLのライブラリオブジェクトをリンクし実行形式のファイルを生成する。ユーザの作成したオブジェクトファイルもリンクすることができるためユーザはC言語を意識することなく多言語でrtKDLのシステムを利用することができる。

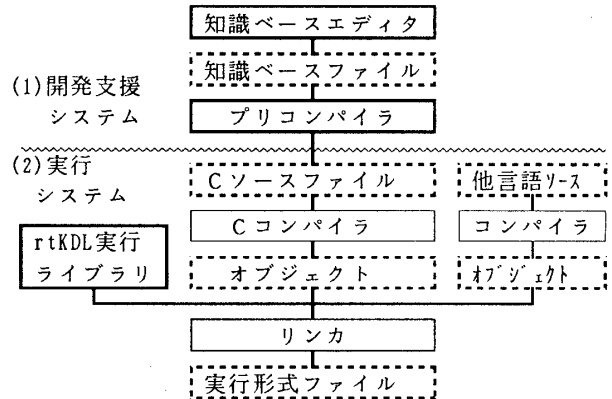


図2. rtKDLシステム構成

6.2 知識ベースの開発手順

rtKDL上で動作するエキスパートシステムを開発するには次のような手順が必要となる。

- ①知識ベースエディタでrtKDLの文法に沿ったソースファイルを作成する。
- ②プリコンパイラによりC言語形式のソースプログラムに変換する。
- ③Cコンパイラによりオブジェクトファイルを生成する。
- ④リンカによりオブジェクトをrtKDL実行時ライブラリやユーザのオブジェクト、ライブラリ等とリンクして実行形式ファイルを生成する。
- ⑤rtKDLの実行システムのデバッグ機能を用いて知識ベースのデバッグを行う。

7. まとめ

監視制御等の実時間制御を要求される分野にエキスパートシステムを適用する上で、Retアルゴリズムの適用や独自の推論制御機構による推論速度の向上により、十分な機能を持つ構築支援ツールを開発できたと考える。今後、分野別に最適な推論システムの開発を進めていく考えである。

参考文献

- [1] 「エキスパートシステム構築支援ツールKDL」-概要-; -推論機構-, 第37回情処全大, 1988.
- [2] 「実時間問題向きシェルrtKDLの開発(2)」-推論機構と知識表現-, 第39回情処全大, 1989.
- [3] Forgy, C.L.: "A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem," Artificial Intelligence, vol.19, 17-37, 1982.