

知識ベース管理システムKappa

3 M-9 -大規模主記憶を利用した全体システム-

金枝上教史

ICOT

横塚実*

*JIPDEC

1. はじめに

われわれはKappaの開発を2つのフェーズに分けておこなっている。現在、最初のフェーズの試作システムの評価をおこない、2つ目のフェーズの全体システムの設計をおこなっている。

試作システムではデータベースとしての機能(小澤)と性能に重点を置き、知識ベースへの足掛かりとなる機能を付加した(宮地、根本)。そして、全体システムでわれわれは2つの目標を持っている。1つはデータベースとしての機能と性能の向上であり、もう1つはKappaの最終目標である知識ベース機能の開発である。

まず性能向上のために数種類の主記憶データベースを構築し、構造を持つデータをより高速に扱う。そして機能の拡張として意味ネットワークの実装と操作コマンドの提供、そして分散機能とより高位のコマンド(河村)の提供をおこなう。意味ネットワークは自然言語処理に必要な知識であり、試作システムの基本モデルの1つにくわえることにした。また知識ベースの機能として、推論エンジンの実装をおこない意味ネットワークを表現する拡張フレーム表現の提供をおこなう。本発表では推論エンジンと主記憶データベースを持つ全体システムの構成と意味ネットワークを構築するための拡張フレーム言語の説明をおこなう。

2. システム構成

全体システムは図1のように推論エンジン(Inference Engine: IE)とデータベース(Database: DB)部から構成される。そしてIEはさらに、Query最適化部と内部DB(Intensional DB: IDB)と実行部よりなり、DBはDB管理部と4種類のDBからなる。以下に各部の概要を説明する。

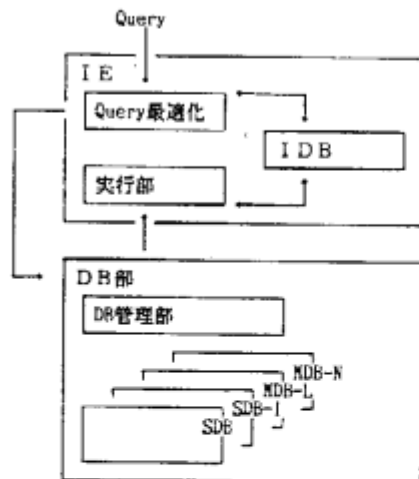


図1. システム構成

[推論エンジン]

IEはユーザからのQueryをDB内の知識を用いてIDBに最適実行の形式に展開する。実行部はIDBのプログラムを順次実行し途中結果を管理する。ただしユーザはQueryの発行前に必要なプログラムの使用宣言をおこなわなければならない。この宣言は「知識源」の単位でおこなわれる(河村)。この推論エンジンの対象として現在ホーン節(河村)、フレーム操作プログラム、CRL(横田)を検討している。

[DB部]

DB部はDB管理部とDBからなる。DBは論理的には原始モデル(小澤)、基本モデル(根本)から構成されるが物理的には以下に説明する4種類のDBから構成される。これらDBは2次記憶および主記憶に存在し、図2のような階層構造をなしている。以下に4種類のDBを説明する。

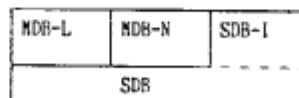


図2. データベースの階層構造

《DB管理部》

4種類のDBに対して資源管理、ユーザ管理をおこない原始モデル、基本モデルのデータ操作を提供する。そして基本モデルでは新たに意味ネットワークの実装をおこないと操作を提供する。

《SDB》

他の3種類のDBの基礎となるDBで全情報が2次記憶上に存在し主記憶はバッファプールとして利用される。さらに各種ユーティリティによりDBの保全が計られている。

《NDB-L》

SDBのデータを主記憶上に展開したDBである。データ操作はすべて主記憶上でおこなわれるので性能が向上する。処理はSDBと同様トランザクション単位におこなわれ、エンドトランザクション時に更新内容をログにはきだすことにより、SDBに遅延更新をおこなう。そしてSDB上では複数のテーブル操作により実現していた基本テーブルに実体を持たせることができ、特に構造型テーブルや結合型テーブルのデータ、そして意味ネットワークデータを主記憶上で直接表わすことにより、これらのデータに対する操作の性能向上が見込まれる。

《NDB-N》

機能はNDB-Lと同じであるが、違いは更新ログがとられないことである。これによりトランザクション単位の処理速度が速くなるが、SDBに更新内容を反映することができなくなる。SDBを持たないKappaやアプリケーションのデバック時に有効である。

《SDB-I》

SDB中のデータを高速検索するためのインデックス、制約情報を主記憶上に持ち、データはSDB中に持つ。トランザクション中の更新内容はエンドトランザクションで逐次

Knowledge Base Management System Kappa
- Total System with Large Main Memory -
Atsushi KANAEGAKI, Minoru YOKOTSUKA*
ICOT *JIPDEC

SDBに反映される。データ量が膨大であったり主記憶容量が乏しく、RDB-Lに格納できない場合にSDBより高速に検索がおこなえる。

3. 意味ネットワーク

全体システムでは自然言語処理に必要な知識として新たに意味ネットワークを構築する。そこでこの節では意味ネットワークの構築のためのフレーム表現について説明する。ここでは、フレームの拡張として『世界』の概念を導入した。その理由は意味ネットワークを構成するノードを同じフレームをもつものの集合としてまとめることにより管理を容易にすることが目的である。ここで使われる言葉とそれぞれの関係を説明する。

《フレーム》

フレームとは、データを格納するためのスキーマを言う。このフレームは図3.aのように継承関係、インスタンス・スロット、操作プログラムにより定義される。ここで“is_a”は(他世界の)インスタンスとの継承関係を表わし、スロット定義はデータの属性を示すスロット名の指定と、そこに格納されるデータのデフォルト値および制約条件を指定することができる。そして操作プログラムにはフレームに対する特有の操作を定義することができる。図3.bにフレームの例をあげる。

```
world < 世界名 >
is_a   < インスタンス名リスト >
instance { (スロット定義) ";" }
< 操作プログラム >
end.
```

図3.a フレーム定義

```
world 生物
instance 飛行能力;
end.
```

図3.b フレーム例

《インスタンス》

世界の構成要素である。インスタンスはフレームにインスタンス名とスロット値を与えることにより世界に登録される。図4.aで生物フレームのインスタンスとして鳥類、哺乳類、魚類の3つを生成し、その結果bに生物世界にインスタンスが登録された様子を示す。

図4.aの()内の各第一要素がインスタンス名で、このインスタンス名は世界内でユニークで各インスタンスは“世界名”インスタンス名”により一意に識別される。そして“=”の後の“あり”、“なし”がスロット値である。スロット値には他のインスタンスを指定することもでき、この時スロット名がインスタンス間の関係を表わす。ただし“is_a”という名前のスロットは他のスロットとは違い継承機能を持つ(フレーム定義の“is_a”とは違う-後述)。

```
生物 ← [鳥類, 飛行能力 := あり],
生物 ← [哺乳類, 飛行能力 := なし],
生物 ← [魚類, 飛行能力 := なし]
```

図4.a インスタンス登録例

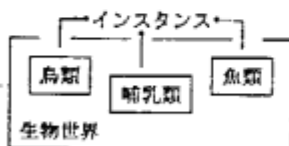


図4.b 生物世界

《世界》

単一フレームのインスタンスの集合。意味ネットワークは複数の『世界』から構成され、インスタンスは個々のノード、スロット名および継承関係はノード間のリンクとなる。

《継承》

2種類の継承のうちの1つは、『他世界』との継承を示し、他世界のインスタンスを上位インスタンスとして指定する。この継承はフレーム定義で指定され、定義された世界の全インスタンスに継承される。そしてもう1つの継承は『自世界』内の継承関係を示し、自世界のインスタンスを上位インスタンスとして指定する。この継承はインスタンス生成時に指定され、指定されたインスタンス間のみおこなわれる。

これらの継承はともに指定されたインスタンスのスロット値とともに継承し、その値はdefault値としての役目を果たす。ただし、『他世界』継承の場合には『下位世界』のフレーム定義において同一名のスロットを定義した場合には、その値は継承されない。

図6および図7は、図5の鳥の定義をもとにした継承の例である。図6は『他世界』継承の例であり生物世界の鳥類インスタンスを継承した鳥世界は飛行能力のデフォルト値を“あり”とする。よって“鷹”の飛行能力は“あり”となる。しかし“ペンギン”はインスタンス登録時に“なし”とすることができる。図7は『自世界』継承の例である。ここでは“ひよこ”が“にわとり”を継承することにより飛行能力として“なし”を継承する。

```
world 鳥
is_a := 生物!鳥類;
instance
is_a;
end.
```

図5. 鳥フレーム定義

```
鳥 ← [鷹];
鳥 ← [ペンギン, 飛行能力 := なし]
```

図6. 他世界継承例

```
鳥 ← [にわとり, 飛行能力 := なし];
鳥 ← [ひよこ, is_a := にわとり]
```

図7. 自世界継承例

4. おわりに

本発表ではKappaの全体システムの構成と意味ネットワークを構築するための拡張フレーム言語の説明をおこなった。現在推論エンジンでの処理方法を検討中である。また拡張フレーム表現は、主に意味ネットワークに必要な機能に主眼をおいてさらに検討をすすめており、今後さらにKappaの非正規形データを扱うことを目指し、Kappaの知識のすべてを記述、管理することのできる1つの知識表現言語としてさらに拡張することを目指している。そして意味ネットワークなどの複雑な構造を持つデータを構築するためのユーザインタフェースも考えている。

参考文献:

- [小澤] 小澤他, “Kappa-試作システムの内部モデル”, 情報処理学会35回全国大会 3M-4, 1987
- [河村] 河村他, “Kappa-利用者インタフェース”, 情報処理学会35回全国大会 3M-10, 1987
- [根本] 根本他, “Kappa-試作システムの基本モデル”, 情報処理学会35回全国大会 3M-5, 1987
- [宮地] 宮地他, “Kappa-非正規形モデルと項の処理”, 情報処理学会35回全国大会 3M-6, 1987
- [横田] 横田, “Kappa-非正規形モデルと演算データベース”, 情報処理学会35回全国大会 3M-8, 1987