

# C++ による汎用フレーム型知識工学環境 ZERO の 2N-6 実現について

小暮慎一

大森康正

上野晴樹

東京電機大学

## 1. はじめに

ZERO は、我々が既に開発した汎用フレーム型知識工学環境であり、Common Lispによって実現され、INTELLITUTORや対象モデルをはじめとした様々な応用に用いられている。現在、ZEROは、Sun, News, NTT ELIS, XEROX 1121等のワークステーション上、及びFACOM M380等汎用コンピュータ上で稼働している。

本システムは、従来のZEROの汎用フレーム型知識工学環境としての長所はそのまま、移植性や既存のシステムとの連結性を高め、更に高速化を図ることを目的にして、ZEROをC++で実現した。したがって、ユーザは、従来と同様の操作方法でシステムを構築することができる。

フレームとC++におけるオブジェクトの間には、フレームがクラス及びインスタンスを操作対象とするのに対して、C++のオブジェクトはインスタンスのみを操作対象とするといった相違点がある。したがって、C++のオブジェクトをそのままフレームの表現に用いることは出来ない。

本稿では、C++のクラスをテンプレートとして用いることでフレームやISA階層を表現する方法、メッセージ変換の管理方法について述べる。

なお、本システムはSunワークステーション上で、Xウィンドウ(X11R5)の環境でSun C++を用いて実現されている。

## 2. 汎用フレーム型知識工学環境 ZERO [1]

ZEROは、複雑で高度な問題解決の為のより汎用な知識表現言語・知識工学研究用環境の提供のために開発された。詳しくは、参考文献[1]を参照されたい。

## 3. C++ による ZERO の実現について

### 3.1 C++によるZEROのシステム構成

ZEROの各モジュール[1]をクラス・オブジェクトによって実現している(図1参照)。を作成し、利用する機能に応じて必要なモジュールのインスタンス・オブジェクトを生成することで、Editor, Activator画面を制御している。

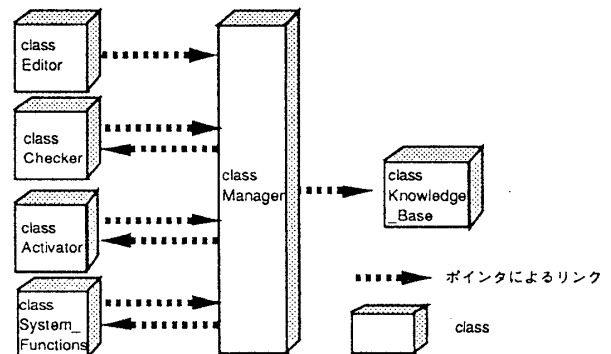


図1: C++によるZEROのシステム構成

### 3.2 C++によるフレーム管理テンプレート

C++のクラス・オブジェクトを用いて、ZEROにおけるフレーム管理の為にテンプレートを準備している(図2参照)。フレームは、このテンプレートを基にインスタンス・オブジェクトによって管理している。

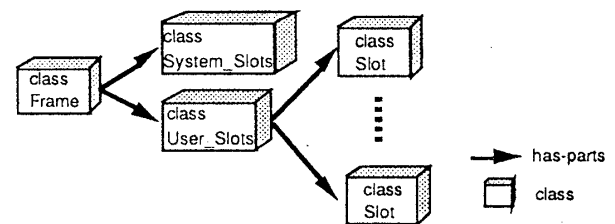


図2: C++によるフレーム表現

### 3.3 知識ベースの管理

ZEROのフレームは、インスタンス・オブジェクトとして管理している。A-kind-ofは、System-Slots

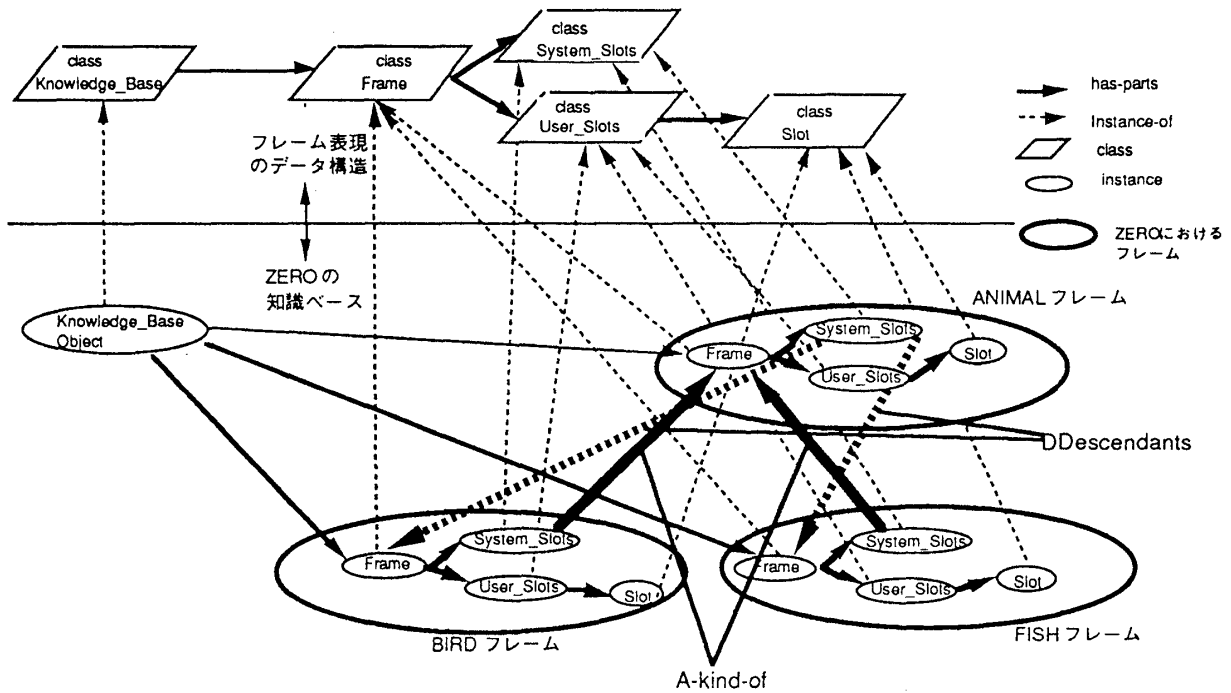


図3：知識ベースの管理

オブジェクトのメンバとして親フレームへのポインタ, DDescendants は子フレームへのポインタとして表現している。これらのポインタによってフレーム間のリンクを実現している (図3参照)。

となった。今後は、本システムの評価を行う。更にユーザ・インタフェースの充実を図る。

3. 4 メッセージ交換

フレーム・システムの推論制御の方法の一つであるメッセージ交換は、インスタンス・オブジェクトによるフレーム表現により、各インスタンス・オブジェクト間のメッセージ交換によって実現している。なお、付加手続きはC, C++によって記述する。

4. 実行例

ZEROの実行例として、エレベータの制御を行うアプリケーションを示した (図4参照)。これはエレベータをフレームとして管理し、フレーム間のメッセージ交換により推論制御を行っている。

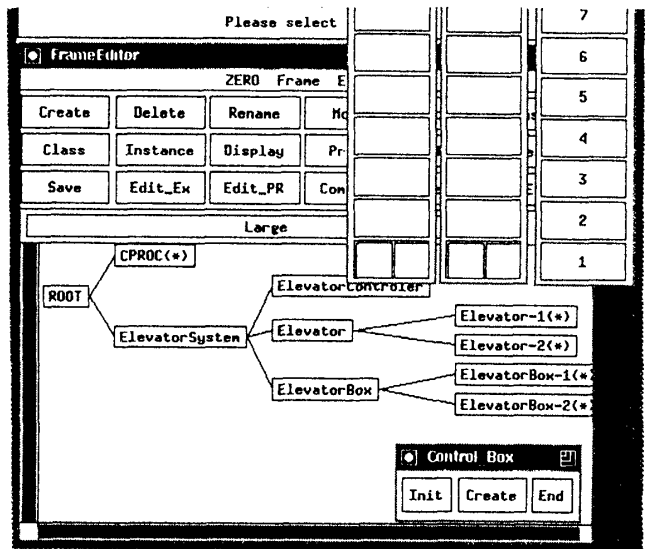


図4：実行例 (エレベータの制御)

[参考文献]

[1] 今井健志, 伊藤浩之, 吉村貞徳, 上野晴樹: 汎用フレーム・システムZERO —その概要とユーザ・インタフェースについて—, 電子通信学会技術研究報告A187-22, 1987

[2] 上野晴樹: 知識工学入門, オーム社

5. おわりに

C++による汎用フレーム型知識工学環境ZEROの実現について述べた。C++のオブジェクトによりフレームを管理し、CやC++を用いた付加手続きの記述が可能となったことで、汎用フレーム型知識工学環境としての長所はそのまま処理の高速化が可能