

言語中心の開発支援環境におけるツール間の連携

6 A E - 8

磯谷 幸太郎 河田 恭郎 前川 守

電気通信大学大学院情報システム学研究所

1 はじめに

現在、我々はより高い柔軟性を重視した新規プログラミング言語、Eunice97 の設計を進めている。この新規言語 Eunice97 の特徴として以下のようなものが挙げられる。

- メタオブジェクト・プロトコル (MOP)[2] を導入。
- 処理系はインタプリタが基本。
- コンカレンシを導入。
- 複数のプラットフォーム上で動作。

ここで、この新規言語での開発を考えたとき、いくつかの開発支援ツールが必要となる。しかし、既存のツールでそのまま使えるものは、テキストエディタなどの非常に汎用のものしかない。

そこで、新たな Eunice97 言語に強力に依存する開発支援環境を構築することを考えた。Eunice97 という言語の特徴を最大限に生かし、個々に作成されたツール群を次のように動作させたい。

- ツールは複数同時実行。
- 同時実行されるツール群は互いに協調動作。
  - コントロール・インテグレーション  
同時実行されているツール同士の連動。
  - データ・インテグレーション  
複数ツール間でのデータの共有。

このように動作する開発支援ツールを作成するにあたり、Eunice97 言語の実装段階で適当なフレームワークを用意しておけば、ツールの作成にかかる手間が軽減できる。

本研究では、以上のような背景下で Eunice97 言語に強く依存した開発支援環境を構築するためのフレームワークを提案する。

なお、ここで想定している開発支援ツールは以下の3種類である。

- 構造エディタ — プログラムを Eunice97 言語の文法に沿って構造的に編集するエディタ
- 視覚的エディタ — Eunice97 言語でのプログラミングを極力視覚的に行なえるようにするエディタ
- 視覚的エグゼキュータ — Eunice97 プログラムの実行を視覚的に表現するツール

2 構成

具体的には、既に Eunice97 が持つ特徴を最大限に利用し、図1のように開発支援環境を構成する。

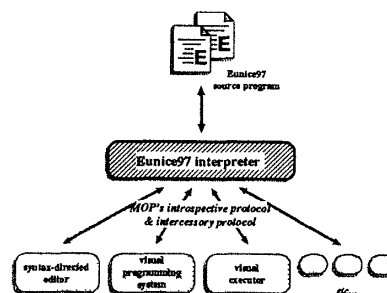


図1: Eunice97 とその支援環境

Eunice97 のソースファイルはすべて Eunice97 インタプリタが管理し、各ツールは必要な処理を Eunice97 インタプリタに依頼する。つまり、編集セッション中のプログラムは常にインタプリタの中に存在し、その状態の問い合わせを MOP のイントロスペクティブ・プロトコルを用いることを行ない、編集操作による状態の変更を MOP のインタセサリ・プロトコルで行なう。

すなわち、ツール作成の際には、Eunice97 インタプリタへの要求手順のみを指定してやればよい。ツールがインタプリタへ依頼する要求メッセージとしては以下のようなものがある。全ての要求はオブジェクト単位の操作要求である。

- オブジェクト参照要求
- オブジェクト変更要求 (作成/削除/更新)
- オブジェクト変更通知要求 (登録要求)
- オブジェクト変更通知取消要求 (取消要求)

A Framework for Tool Integration in A Language-Centered Environment

Kotaro Isogai, Yasuro Kawata and Mamoru Maekawa  
Graduate school of Information Systems, University of Electro-Communications  
1-5-1 Chofugaoka, Chofu, Tokyo, 182, Japan

### 3 コントロール・インテグレーション

コントロール・インテグレーションの実現はオブジェクト変更通知要求を使って行なう。変更通知要求はオブジェクトに変更があったときの通知を依頼する要求である。あるオブジェクトの変更を知らされたときの動作をツール内で決めておけば、他ツールでのそのオブジェクトの変更動作との連携が実現できる。

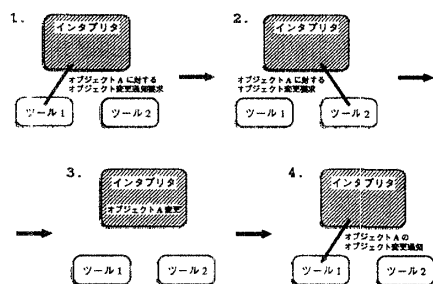


図 2: ツール間連動の実現

図2の例で説明する。まずツール1がインタプリタにオブジェクト A に対する変更通知要求を出しておく(図2-1)。その後、ツール2の操作によってオブジェクト A への変更要求が出される(図2-2)。すると、この要求を受けとったインタプリタは、ツール2の変更要求にしたがって、プログラム中のオブジェクト A に変更を加え(図2-3)、つづいて、ツール1へオブジェクト A に変更があったことを通知する(図2-4)。

このようにして、ツール1がオブジェクト A の変更通知を受けとったときの動作を決めておけば、この2つのツール間の連動が実現できる。

### 4 アノテーション機能

Eunice97 言語は任意のオブジェクトに対して別のオブジェクトをアタッチできるという機能を持つ。

例えば、あるオブジェクトに対して、プレーンテキスト、HTML、ビットマップなどのデータをアタッチするというようなこともできる。このように、任意のオブジェクトにアタッチされたオブジェクトをアノテーション・オブジェクトとよぶ。

アタッチされるアノテーション・オブジェクトには、特に「役割(role)」という属性が与えられる。これは、そのオブジェクトにアタッチされたそのアノテーションが何に使われるものなのかという一種のキーワードであり、同一オブジェクトにアタッチされ

る同じ型のアノテーションについて、この属性値の重複は許されない(図3)。

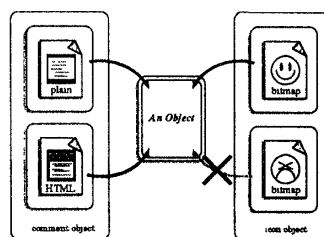


図 3: Eunice97 のアノテーション機能

### 5 データ・インテグレーション

アノテーション機能を利用して、ツール自身が必要な型のアノテーション・オブジェクトを付加しておけば、単一レジスタ上で各々が必要な型でオブジェクトを参照することができ、また、他のツールが作成したアノテーションを参照することも同様に可能である。各ツールはあるオブジェクトのアノテーションを参照したいとき、アノテーションの役割と型を指定することにより、特定のアノテーション・オブジェクトを一意に選択することができる。

このようにして、オブジェクト単位でのデータの共有が実現できる。

### 6 結論

本稿では、柔軟性を求める新規言語 Eunice97 における開発支援環境を構築するにあたっての構成を提案した。さらには新規言語 Eunice97 のもつ柔軟性を可能な限り利用し、コントロール・インテグレーション、データ・インテグレーションを実現する概念的な枠組を述べた。しかし、細部にはまだ問題点も多く、設計段階でさらなる構想が必要と思われる。この枠組を完成させ次第、実際に開発支援ツールの作成に取り組み、その成果を評価するつもりである。

### 参考文献

- [1] Ward Cunningham. Model-view-controller. Technical report, Tektronix, 1985. (抄訳) ソニー・テクニクス(株) AI 営業部, 「Smaltalk-80 によるアプリケーション・プログラムの作り方」, bit, vol. 18, no. 4, pp. 379-396, 共立出版, 1986 年 4 月.
- [2] Gregor Kiczales, Jim des Rivières, and Daniel G. Bobrow. *The Art of the Metaobject Protocol*. MIT Press, 1991.