

新規言語 Eunice97 における開発支援環境フレームワーク

4 C - 4

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科

1 はじめに

現在、我々が設計、開発を進める新規プログラミング言語 Eunice97 は以下のような特徴を持つ。

- メタオブジェクト・プロトコル (MOP) [1] による柔軟な拡張性
- 処理系はインタプリタを基本
- コンカレンシー
- 複数のプラットフォーム上で動作

この新規言語は非常にユニークで、従来のプログラミング言語とは大きく異なる。そのため、この言語での開発支援ツールについては、例えばテキストエディタのような非常に汎用なもの以外は、新規に作成する以外はない。言語の特徴を最大限に生かし、個々に作成される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
- 複数ツール間の連携
 - 複数ツール間で可能な限りデータを共有
 - ツール同士の動作の連動

このように動作するツールを作成するにあたって、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。なお、現在考えているツールは以下の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

本稿では、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の中から、データ統合の方法とその拡張の一部を紹介する。

2 基本構成

具体的に、既に Eunice97 が持つ特徴を最大限に利用し、図1のような構成でツール群を動作させる。

A Tool Integration Framework for Eunice97
Kotaro Isogai, Yasuro Kawata and Mamoru Maekawa
Graduate School of Information Systems, University of Electro-Communications
1-5-1 Chofugaoka, Chofu-shi, Tokyo, 182-8585, Japan

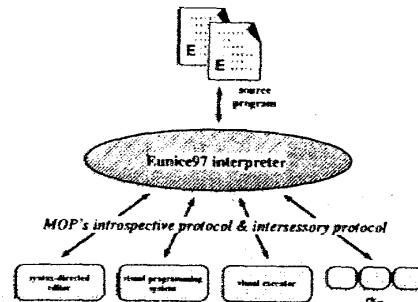


図1 Eunice97 とその支援環境

ソースファイルはすべて Eunice97 インタプリタが管理し、各ツールが必要な処理はインタプリタに依頼することによりインタプリタが全て行なう。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中に存在し、その状態の問い合わせや編集操作による状態の変更は言語の処理系がもつ MOP の機構により行われる。

ツール作成の際には、Eunice97 インタプリタへの依頼手順のみを指定してやればよいことになる。

3 アノテーション機能

Eunice97 言語には、任意のオブジェクトに、アノテーション (注釈) と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たせてやる。主にツールのみが使う、ソースプログラムとは無関係の情報をソース中のオブジェクトに持たせてやるための仕組みである。

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、アタッチされたオブジェクトを特にアノテーション・オブジェクトとよぶ。

なお、ここでのアタッチメントとは役割 (role) を表す属性を持ったリンクによる関連付けのことである。その属性はオブジェクトにアタッチされた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となるものであり、同一オブジェクトにアタッチされる同じフォーマットのアノテーション・オブジェクトについて、この属性値の重複したアタッチメントは許されない (図2)。

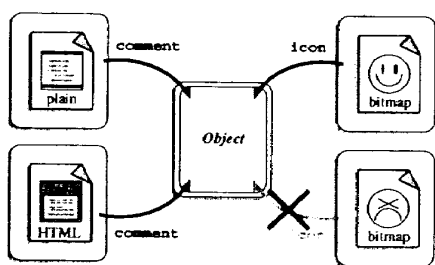


図 2 アノテーション機能

4 データ統合の実現

アノテーション機能を利用して、ツール自身が必要なアノテーション・オブジェクトを付加しておけば、各々が必要な形式でオブジェクトを参照することができる。また、オブジェクトはインタプリタが一元管理しているため、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容易に可能である。各ツールは、対象オブジェクトに役割とその形式を通知することにより、特定のアノテーション・オブジェクトを一意に参照することができる。

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。

5 アノテーションのフォーマット変換

同一内容のアノテーション・オブジェクトであるにも関わらず、ツールによって使用するフォーマットのみが異なる場合がある。その場合、既に存在するフォーマットのアノテーション・オブジェクトを自動的に変換して参照させることになる(図3)。

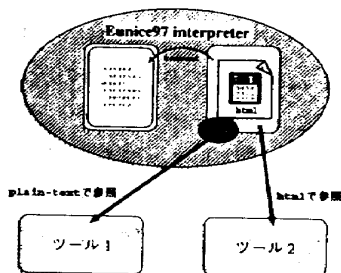


図 3 フォーマット変換による参照

6 ツールの国際化サポート

現在、アノテーション・オブジェクトとして主に想定しているのは構造エディタや視覚的エディタで用いられると思われる、コメントやアイコンである。これ

らはいずれも国際色豊かなものであり、それぞれ地域毎に表現方法が異なる。この点に関して、Eunice97ではツールのロカールによってアノテーション・オブジェクトを使い分けることで国際化を実現する。同一の役割で用いられる同一フォーマットのアノテーション・オブジェクトでもロカールが異なればアタッチ可能である。

7 関連研究

既存の汎用 CASE 統合環境フレームワークとして、FIELD プログラミング環境 [2] がある。この FIELD 環境の Broadcast Message Server (BMS) の機構をお手本に、Sun の ToolTalk や HewlettPackard の Soft-Bench などが生まれた。

この BMS は、本研究と目指すところが重複しているが、既存の言語での制御統合にその主眼点がある。本研究では、Eunice97 という新規言語のひとつのフレームワークとしてツール統合の機構を実装することにより、開発支援ツールの各処理、制御を言語処理系にすべて集中し、アノテーション機能の導入に示されるようにデータまでも処理系内部で統合を図るというアプローチをとっている点で大きく異なる。

8 結論

本稿では、新規言語 Eunice97 における開発支援環境の構成を簡単に紹介した。そして、言語の特徴を生かしたデータ統合を実現する方法とその拡張を概念的な面から述べた。

紙面の都合上、データ統合に関する一部を紹介するに留まったが、この他にツール間連動を実現する方法や、その他諸々が既に定式化されている。またこれらの枠組を基に作成したツールも存在する。

参考文献

- [1] Gregor Kiczales, Jim des Rivières, and Daniel G. Bobrow. *The Art of the Metaobject Protocol*. MIT Press, 1991.
- [2] Steven P. Reiss. *The FIELD Programming Environment: A Friendly Integrated Environment for Learning and Development*, volume 298 of *Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, 1995.