

HiLISP統合型プログラミング環境

6S-8

青島利久 武市宣之 ((株)日立製作所 中央研究所)

泉達也 (日立東北ソフトウェア(株))

山本栄次(日立ソフトウェアエンジニアリング(株))

1.はじめに

我々は、高速なCommon LISPの処理系HiLISP[1]を開発し、その上にエディタを中心とするプログラミング支援環境[2]を開発してきた。

今回、日立ワークステーション2050上での効率的なプログラム開発を支援するために、既存プログラムの再利用や動作解析ツールを統合化した開発環境を試作したので報告する。

2.プログラム作成における課題

LISPのプログラム開発では、コーディング・実行・動作解析・修正のサイクルを何度も繰り返す逐次改良型のプログラミングが多く行われている。このような開発環境では、以下に列挙する課題がある。

- (1) プログラムの大規模化に伴い、プログラムのコーディング・解析・修正中にプログラムの全体構造を把握する手段がなかった。
- (2) プログラムの動作が適切かどうか解析する際に、大量に生成される関数の引数や戻り値のデータを効率的に参照することができなかった。
- (3) 関数内での変数の更新状況の調査に手数を要した。

3.統合環境の全体構成

図1に、その全体構成を示す。既存のエディタ[2]、インタプリタ、コンパイラに、新たにオーバビューアとアナライザ(実行履歴検索)とビジュアルステップバを開発した。各ツールは、オーバビューアを介して機能が統合化されている。

Integrated Programming Environment For HiLISP
Toshihisa AOSHIMA(1) Nobuyuki TAKEICHI(1)
Tatsuya IZUMI(2) Eiji YAMAMOTO(3)
(1)Central Research Laboratory,Hitachi Ltd.
(2)Hitachi Tohoku Software Co.,Ltd.
(3)Hitachi Software Engineering Co.,Ltd.

4.特長機能

(1)オーバビューア

ディスクにあるファイル名の一覧、メモリ環境にある関数名の一覧、ソースプログラム・レベルでの関数の呼出し関係を表す関数系統図の生成・表示機能を有している。

また、オーバビューアで表示された対象を選択指示した後、指定のファイルの読み込みや、エディタやコンパイラ等の支援機能呼び出すことができるようになっている。

(2)アナライザ(階層実行履歴検索)

プログラム実行時に、実行した関数の呼出し関係と引数・戻り値の実行履歴データを取得する。そしてプログラムの中断あるいは実行完了後、アナライザの起動により関数系統図を表示する。更にその関数系統図上で選択した関数の実行履歴データを別のウィンドウに表示する。

また、表示した関数の実行履歴データに対して、指定の関数から直接呼ばれた下位の関数の実行履歴データを展開表示したり、逆に下位の関数の実行履歴データを一括して省略表示することを、各関数の実行履歴データの先頭に付加して表示されるアイコン(⇔▼▲)をマウスピクすることにより可能とした。この階層検索機能により大量の実行履歴データの中から、必要な階層レベルのデータを効率よく参照でき、プログラムの誤りを含む関数の特定が短時間で可能となった。図2に、実行の表示例を示す。

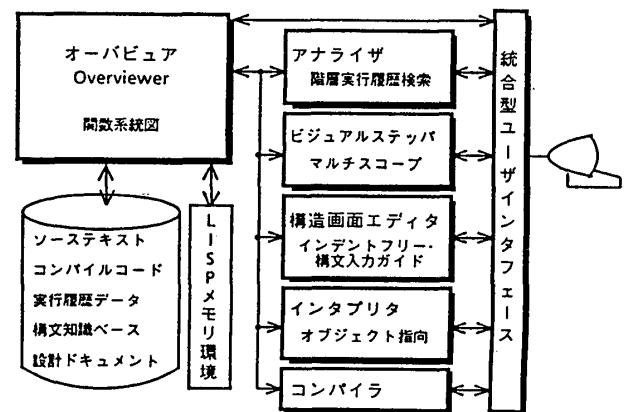


図1. 統合型プログラミング環境の構成

(3) ビジュアル・ステッパ

本ビジュアル・ステッパは、実行プログラムの全体構造を表す関数系統図と、現在実行中の関数のソーステキストの両方を並べて表示し、それぞれに現在位置を識別表示するようにした。更に別のウィンドウには、現在実行中の関数における引数の値、ローカル変数の値、大域変数の参照・更新値をレポートする。このようにプログラムの実行状況を複数の視点から観察できるようにした(マルチスコープ)。図3にその表示例を示す。ここに表示される関数系統図は、ステッパ起動時に、ソーステキスト上から直接抽出される静的な呼出し関係の全体図であり、系統図にない関数が動的に呼び出されたときは、その時点で静的な関数系統図に新しい枝を追加する方式とした。

5. 効果

オーバビューやアナライザ、ビジュアル・ステッパでの関数系統図の表示によって、任意の時点でのプログラムの全体構造の把握が容易になったことにより、プログラムの再利用や解析におけるプログラマの負担が軽減している。これらの機能はユーザから好評を得ている。

図4は、プログラムの動作解析、修正、確認実行における1サイクル当たりのプログラマの手数を、従来の環境と本統合環境について帯グラフで比較したものである。今回新規に開発した統合環境のツールにより、従来最も手間のかかっていた動作解析・誤り箇所検索部分が大幅に短縮され、解析・修正・確認実行のサイクルに対するユーザの負荷が従来の2/5になった。

6. まとめ

開発対象のプログラムの構造を概観するオーバビュー、関数の実行履歴を階層検索するアナライザ、プログラムの実行状況をマルチスコープで監視するビジュアル・ステッパを特長とする統合型プログラミング環境を開発し、その効果を算定した。

当プログラミング環境は、現在当社中央研究所内のLISPユーザの間で試用している。また、これと並行して実行履歴取得による処理性能の低下や所要メモリの増大等の改善を図っている。また、オブジェクト指向CLOSのクラスやメソッドのオーバビュー機能の搭載などの機能拡張も行っている。

参考文献

- [1] 武市 他: 「知識処理用言語HiLISPの高速化方式」 日立評論 vol.69 1987.3
- [2] 黒須 他: 「HiLISPプログラミング支援環境」 情報処理学会第34回全国大会, 1987

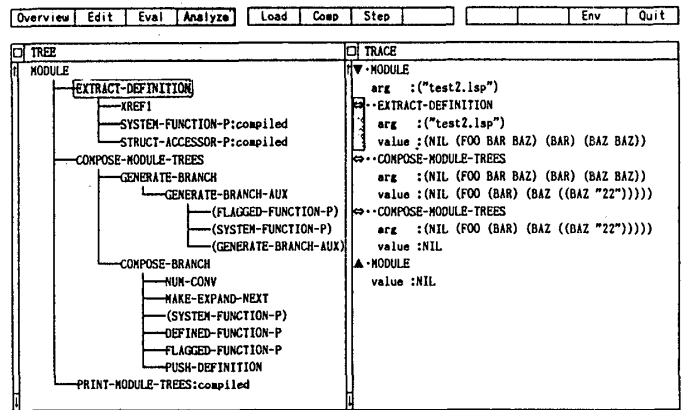


図2. 階層実行履歴検索画面例

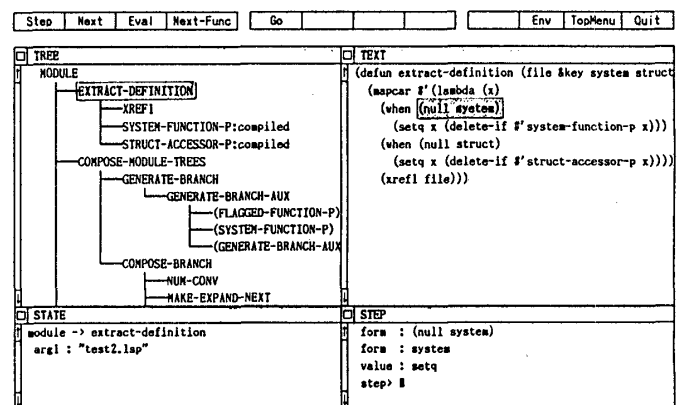


図3. ビジュアル・ステッパ実行画面例

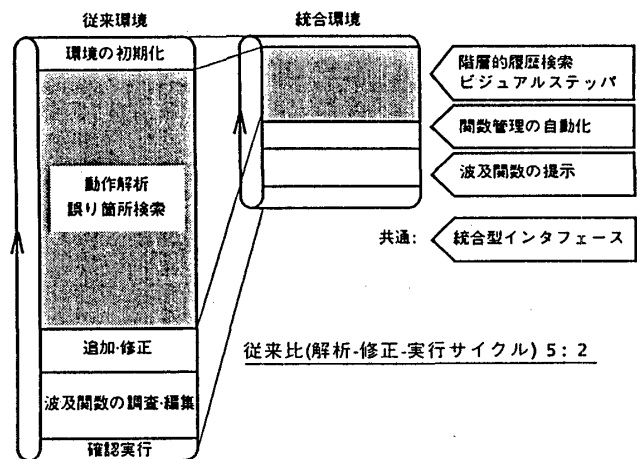


図4. プログラム作成におけるユーザ負荷の比較