Fortran 構造ツリー表示システムの Eclipse 上への実装

大市知未 [1]，横川三津夫 [1]，寺井優晃 [2]，南一生 [2]

プログラムのチューニングや並列化作業を行う前段階として，プログラム構造を把握する必要がある．特に，プログラム中のループ，分岐，プロシージャ呼び出しの構造を理解出来れば，プログラム全体の構造を把握することが容易となる．本研究では，FORTRAN 77 及び Fortran 90 で書かれたプログラムについて，プログラム構造把握の作業効率向上を目的とし，ループ，分岐，プロシージャ呼び出しをプログラムから抽出し，ツリー状に可視化する支援ツール (STView) をフリーの統合開発環境 Eclipse 上に構築した．

1 はじめに

既存のプログラムをチューニング，並列化・高速化する作業では，まず初めにプログラムの構造を把握しなければならない．チューニングを必要とするプログラムは数値シミュレーションなど Fortran で書かれた大規模なものが多く，プログラム構造を把握するだけでも多くの労力を要する．しかし，Fortran のプログラム構造を把握するための無償の支援ツールはほとんど無い．数少ない無償の支援ツールとして，理化学研究所が公開している K-scope[1] が挙げられる．しかし，K-scope は導入するために独自の環境を必要とするので導入の際に時間が掛かることがある．そこで今回，K-scope のように計算コストは細かく計算することは出来ないが，独自の環境ではなく広く知られている統合開発環境 Eclipse[2] 上にプラグインとして構築することにより，容易に使えることに焦点を当てたプログラム構造の可視化を行う支援ツール「STView」を開発した．

2 ツール概要

STView は Java で構築しており，Eclipse 環境にプラグインとして追加することにより使用することが出来る．インターフェースは図 1 のビューの形式を採用しており，ビュー内にあるツリー作成ボタンを押すことにより構造ツリーの作成を行う．処理の流れはプロジェクトフォルダ内のソースファイルを静的に解析，構文に対応するノードクラスを作成，プロシージャ毎にツリーブロックを作成する．最終段階として，program 構文のツリーノードを中心とし，プロシージャ呼び出しのツリーノードの部分には対応するプロシージャのツリーブロックを子ノードとすることで，プログラム全体の構造ツリーを作成し，可視化を行う．また，Fortran 構文全てのノードクラスを作成する訳ではなく，構造把握に重要となるループ，分岐，プロシージャ呼び出しにあたる Do，If，Call，Function 構文をツリー構造状に可視化している．

本ツールは Eclipse のプラグインなのでエディタ画面でソースファイルのプログラムを変更してもツリーの再構築を簡単に行うことが出来る．また，ノードクラスにはソースファイルの情報も含まれているので，STView のツリーノードをダブルクリックするとエディタ画面に対応するソースファイルの行でハイライトされ，該当する箇所を容易に表示出来る．

STView では，可視化後に構造把握をより容易とするためにフィルター機能を実装している．フィルター機能ではユーザが正規表現を用いて検索することにより，ルートノード (program 構文のツリーノード) から正規表現に対応するノードまでを図 1 の部分に表示させ，他のツリーノードを非表示とすることが出来る．逆に，正規表現に対応するツリーノードを非表示にすることも出来る．この機能を使い，必要な部分のみを可視化することで，プログラム構造の把握作業がより効率的に行うことが出来る．更に検索機能も実装しており，こちらの機能はフィルター機能と同様にユーザが正規表現を用いて検索を行うが，ツリーノードの表示/非表示を変更せずに入力した正規表現に対応するツリーノードがハイライトされる．正規表現に対応するツリーノードが複数ある場合，全てのツリーノードがハイライトされるので，検索機能を用いることで同じサブルーチンがプログラムのどの場所で呼ばれているのか等が容易に知ることが出来る．

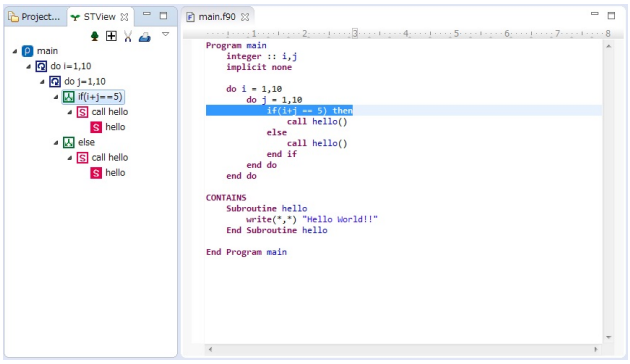


図 1: STView インターフェース

3 まとめ

STView の動作検証を行うために，EigenExa[3]，DNS[4]，姫野ベンチマーク [5] の Fortran 版プログラム (himenoBMTxp_m_omp.f と himenoBMTxpr.f90) の 4 つのプログラムについて可視化を行ってみたところ，問題無く動作した．なお、ソースファイルに対するパーサ部分は現在自作したものを使用しているが，今後は Open Fortran Parser[6] を組み込むことで解析精度の向上を行う予定である．

本研究の一部は，計算科学振興財団の研究教育拠点 (COE) 形成推進事業「超並列プログラムの開発・利用環境技術の展開と人材育成」の支援を受けています．

参考文献

[1] K-scope (http://www.aics.riken.jp/ungi/soft/kscope/).

[2] Eclipse (https://www.eclipse.org).

[3] 高性能固有値ソルバ EigenExa (http://www.aics.riken.jp/labs/lpnctrt/EigenExa.html).

[4] M. Yokokawa, K. Itakura, A. Uno, T. Ishihara, and Y. Kaneda, “16.4-Tflops Direct Numerical Simulation of Turbulence by a Fourier Spectral Method on the Earth Simulator,” *Proceedings of the 2002 ACM/IEEE conference on Supercomputing*, pp. 1–17 (2002).

[5] 姫野ベンチマークプログラム (http://accc.riken.jp/2461.htm).

[6] Open Fortran Parser (http://fortran-parser.sourceforge .net).

[1] 神戸大学
Kobe University
[2] 理化学研究所　計算科学研究機構
RIKEN AICS

ⓒ 2015 Information Processing Society of Japan