6N-2

実行時エラーのプログラム相談システム『Consult:R』

佐藤眞木彦*1, 斉藤　哲*1, 岡本匡人*1, 小林正和*1
大西　淳*2,　島崎眞昭*3
(　*1富士通株式会社,　*2京都大学,　*3九州大学　)

1. はじめに

我々は、「FORTRAN プログラムに関するプログラム相談の自動化」という研究プロジェクトConsult を進めている。現在は、コンパイルエラーを相談の対象にしたConsult:Cと実行時エラーを対象にしたConsult:R の二つを並行して進めている。Consult:C については情報処理学会の研究会や全国大会等で発表してきた〔1,2〕ので、ここではConsult:R について紹介する。

実行はできるが答が間違っているというようなアルゴリズムの誤りの対処を扱うことが非常に困難であるため、Consult:R の相談の対象のエラーは実行時に異常終了したものと実行時エラーが検出されたものに限定している。又エラーによっては単に相談を行うだけでなく、コンパイラのデバッグオプションの使用を推めるなど、実際のプログラム相談に則した対応が取れるようになっている。

2. Consult:Rの概要

Consult:Rの相談の対象としたエラーは、ABEND コードとエラーコードが検出されるものに限定した。富士通のFORTRAN の実行時のエラーコードは全体で 200程度あるが現在は手法の有効性の検証を目的としたプロトタイプ作成を主眼においており、全てに対処するのではなく、その中でも資料〔3〕や実際に初心者の作成したプログラムの実行結果を参考にして、発生頻度の高いと思われる20程度を選択しその対処をすることにした。

FORTRAN のエラーを初心者が修正を試みる際に次のような点が障害になっていると思われる。

1. エラーコードやそのメッセージの意味が判りにくい。
2. エラーコードのみからは、どのように修正すべきか判らない場合が多い。
3. エラーの真の原因となる場所とは異なる所でエラーメッセージが出る場合がある。
4. 通常デバックオプションはつけないために適切な警告がなく、原因が判りにくい。

以前研究された相談手法〔4〕ではエラーコードごとに原因を提示するにすぎず、上記の問題にうまく対処できなかった。我々はソースプログラムを解析することによって、エラーコードを生じさせている真の原因を見出す手法を検討した。手法の特徴を次に示す。

① エラーコードを手掛かりにソースを調べることによってエラー原因の推論を行う。
② エラーコードに対するエラー原因を予め想定して用意しておく。
③ 解析してもエラー原因が特定出来ない時は、デバッグオプションとDATAON（入出力の際のデータを書きだす機能）等を付加して再コンパイルと再実行を行いその結果をもとに解析する。

本手法は従来のものより次の点で優れていると考えている。

i) より正確にエラー原因が特定する。
ii) 他のエラーが原因で生じたエラー（波及エラーと呼ぶ）についても対処する。
iii) 初心者には使用が困難なデバッグ機能を必要な場合は自動的に付加して解析を支援することができる。
iv) プログラム単位を超えたエラーに対処できる。

3. 相談システムの構成

図1は、相談システムの流れを示したものである。現在はエラー原因の推論アルゴリズムの設計を終え、システム全体の作成にとりかかっている。

相談システムの起動により、対象ソースのデータセット名を指定する。ここでファイルのアロケート情報を見ることも出来る。実行環境はシステムを起動する前の環境をそのまま使用出来るが、システムの内部からファイルのアロケートを行う事により実行環境を設定する事も出来るようにする予定である。

環境が設定されソース名が指定されるとシステムは自動的にコンパイル・実行を行い、その結果を指定したファイルに出力する。この時コンパイルオプションを利用者が指定する事も出来る。実行が終了（異常終了も含む）するとソース・コンパイル情報（XREF等）・実行結果（エラーコード）を取込み内部データとする。

次にエラーコードの選択画面に移り、選択されたエラー

PROGRAM CONSULTATION SYSTEM FOR RUN-TIME ERRORS , CONSULT:R.

Makihiko SATO*1, Akira SAITO*1, Masato OKAMOTO*1, Masakazu KOBAYASHI*1
Atsushi　OHNISHI*2, Masaaki SHIMASAKI*3
(　*1FUJITSU LTD.　*2KYOTO UNIV.　*3KYUSHU　UNIV.　)

コードをトリガにしてエラー原因の推論に移る。この時にデバッグオプションを付ける必要があると判断されると、利用者に確認を行ってデバッグオプション付で再コンパイル・再実行し、改めて相談を行う。

尚、ループに陥り相談システムに制御が戻らずに異常終了することを防ぐため、実行時間はCPU-TIMEが10秒に制限されている。

4. エラー原因の推論

エラー原因の推論に当たっては、各エラーコード毎に考えられるエラーの原因とその判定方法をあらかじめ用意しそれを元にエラー原因の推論を行う。

エラー原因の推論は、前に述べたように単にエラーコードと原因の関係のみでなく、ソース、コンパイル情報、実行結果などを参照して進められる。また、場合によってはコンパイラのデバッグオプションを付加して再実行し、新たにデバック情報を採取した上で解析する。更に入出力関係のエラーでは、実行時オプションのDATAONを付加して再実行される。

図2はエラーコード:JZL186I-Wとその対処の例である。このエラーコードはデバッグオプションUNDEF を指定した時に検出の対象となる。このエラーコードの相談過程を以下に示す。

①コンパイル情報をみてJZK524I-I(値が未定義) が出ているかをチェックする。検出されていればJZK524I-I に対するエラーの対処と同様の処理をする〔1〕。これは、文の後半にGOTO文があるか、名前に関する波及エラーであるか、タイプミスの為に類似した文字列があるか等のチェックである。

②これが無ければnameの変数名に類似した変数名を使用していないかのチェックに移る。これはこのエラーがタイプミス等による間違いから生じる事を想定しているためである。類似した変数名があればそれを示しタイプミスであるかを確認する。

③次に利用者がゼロクリアを期待して使用しているかを聞く。利用者の回答が否定的であれば、アルゴリズムの間違いで値が代入されていない可能性を示唆する。
肯定的な回答であれば、初期値を設定するように指示する。

5. おわりに

今秋にはプロトタイプの作成を終了し、その評価を進める予定である。

現在、プロジェクトはConsult:C,Consult:R と分かれているが、将来的にはこの2つを合わせて総合的なプログラム相談システムに発展させたいと考えている。

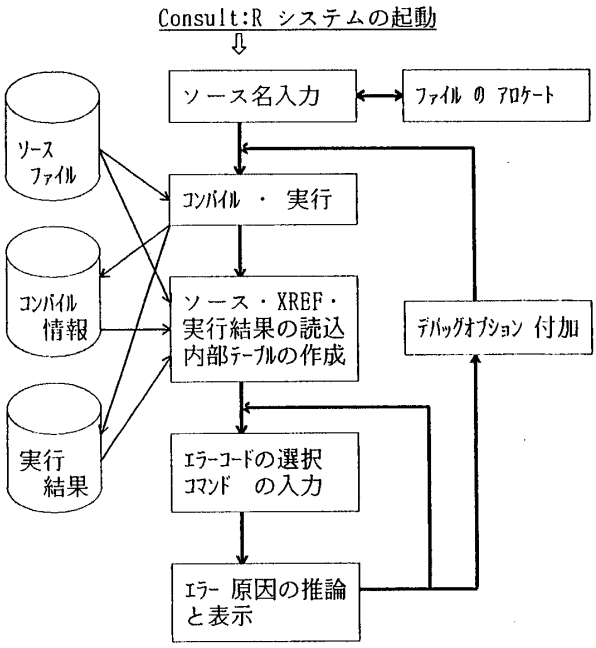

図1　相談システムの流れ

JZL186I-W isn : name HAS AN UNDEFINED VALUE.
(name の値が未定義である。)

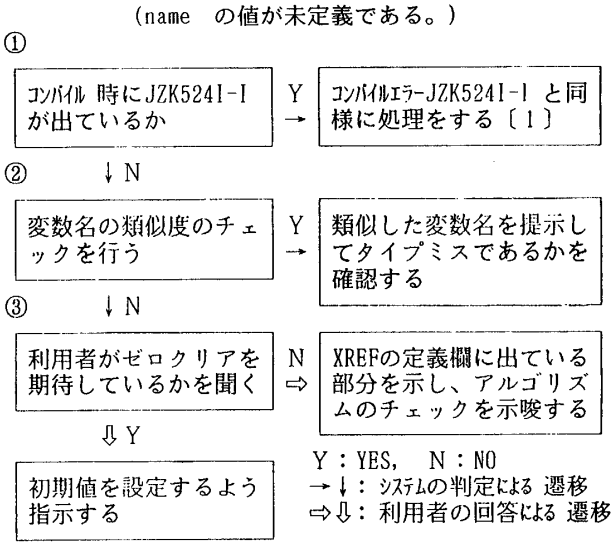

図2　エラーコードとその対処の例

参考文献

〔1〕大西他：ＣＯＮＳＵＬＴ／Ｃ：コンパイル時のエラーに関するプログラム相談手法／システム，情報処理学会ソフトウェア工学研究会64－24 (1989)

〔2〕斉藤他：Consult (コンパイル編) のシステム化について，情報処理学会第38回全国大会3N-8 (1989)

〔3〕FORTRAN プログラムデバックの手引　第2版，東京大学大型計算機センター(1986)

〔4〕田中　一：プログラム相談の機械化報告書，文部省科学研究費補助金による特別研究「広域大量情報の高次処理」(1976)