

コンパイラエラーのプログラム相談システム『Consult:C』の評価

6N-1

斉藤 哲<sup>\*1</sup>, 佐藤真木彦<sup>\*1</sup>, 岡本匡人<sup>\*1</sup>, 小林正和<sup>\*1</sup>  
 大西 淳<sup>\*2</sup>, 島崎眞昭<sup>\*3</sup>  
 (\*<sup>1</sup>富士通株式会社, \*<sup>2</sup>京都大学, \*<sup>3</sup>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』についての研究プロジェクトを行っている [5]。その中で、FORTRAN プログラムのコンパイラエラーを対象としたプログラム相談システムを『Consult:C』と呼んでいる。

この『Consult:C』のプロトタイプを作成し [4]、その後、一般利用者に試用していただき、その結果を評価したので報告する。また、プロトタイプの問題点を改善した実用版システムを作成中であり、それについての概説も報告する。

2. 『Consult:C』の概説

『Consult:C』はFORTRAN プログラムのコンパイラエラーのプログラム相談を自動化することを目標として作成したシステムであり、以下の特徴をもつ。

- ① エラー原因の推論をソースをもとにして行う。
- ② ソースプログラムの解析をコンパイラと異なる観点で行う。(空白をデリミタとする、カラムをずらして解析する、等)
- ③ 推論駆動のトリガとしてエラーコードを用いる。
- ④ 他のエラーにより発生するエラー (波及エラー) に対処できる。
- ⑤ 推論結果を利用者に問い合わせる。
- ⑥ コンパイル時のオプションについては本システムが予め用意したものを指定し、利用者が指定することはできない。

図1に『Consult:C』のプロトタイプの実行過程を示す。プロトタイプは全体制御部と推論部にPrologを用い、利用者インターフェイスにCOBOLを用いて実現されている。

3. プロトタイプの評価

『Consult:C』のプロトタイプの試用を富士通電算機専門学院に依頼し、その結果を評価したので報告する。

プロトタイプでは、利用者がプログラム相談を行うと自動的に各利用者単位に以下の項目について相談過程の記録を行う。この内、(c)(d)(e)は実際の画面イメージをそのままファイルに記録している。

- (a) 相談対象のソースプログラム
- (b) (a)より発生したコンパイラエラー
- (c) 相談を行ったエラーコード
- (d) (c)に対する『Consult:C』からの推論結果
- (e) (d)に対する利用者の応答

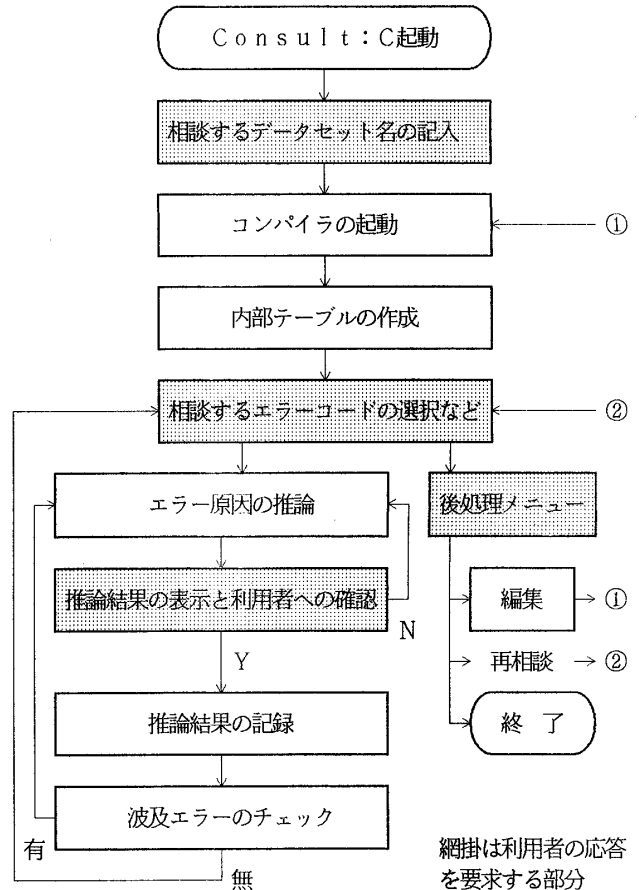


図1 Consult:Cの実行過程

表1 相談結果の分類 (単位:件)

対処可能	正確な相談結果 (波及エラーに対処)	34 (9)
	不十分な相談結果	20
	見当はずれな相談結果	22
未 対 処		86
合 計		162

Estimation of program consultation system for compile-time errors(Consult:C)

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

プロトタイプでの試用結果で相談が行われた件数（項目(c)の合計）は162件であった。この各件を項目(d)(e)に記録されている画面イメージをプロジェクトチームで評価し、相談結果として正しい推論を行っているかを集計したものを表1に示す。

この結果からプロトタイプでは発生したエラーのほぼ半数しか相談対象としていないことが判明した。プロトタイプで採用しているエラーコードは京都大学大型計算機センターで発生した頻度の高い30種であるが、試用を依頼した富士通電算機専門学院の場合はこれらのエラーでは十分に対応しているとは言えない。

対処可能なエラーコードの中で『不十分な相談結果』や『見当はずれな相談結果』が多かったのは、『デリミタの誤り』『FORMAT文の誤り』『IF文の誤り』であった。また、『分岐文の直後にも関わらず文番号がない』や『END文がない』といった比較的簡単と思われるエラーについても対処できていないものもあった。

図2-1、図2-2に正確な相談結果を出力したものと見当はずれな相談結果を出力したものを具体的に示す。

図2-1は行15が未宣言であるというエラー（JZK379I-S：この文は英字名を配列名又は文字型のように引用している）について相談を行っている例であるが、本システムはこのエラーが行6のエラー（JZK424I-S：これはFORTRANの文とは思えない）で行6自体が無効となっているための波及エラーであることを指摘している。また、図2-2は未宣言の配列変数があったために発生したエラーであるが文関数定義の誤りの疑いを先に行っているため、見当はずれの指摘を行っている。その後、利用者が『No』を入力すると正しい指摘を行う。

#### 4. おわりに

今回評価した試用結果から以下の考察を得た。

- (1) プロトタイプで対処できるエラーコードは京都大学大型計算機センターで発生したFORTRANプログラムのコンパイルエラーの頻度の高い30種であったが、試用者のエラーの傾向とはかなり異なったものであった。より多くの試用結果を集め、実用版システムで対処したい。
- (2) プロトタイプではキーワードのスペルミスと変数名のスペルミスが重なったような複合エラーには対処できない。この点においては実用版システムで検討したい。
- (3) 利用者には『間違っ癖』があるように思われる。CAIで使われる利用者モデルを導入して推論順序を変更する仕組みを検討したい。

以上を考慮し現在作成中である『Consult:C』の実用版は以下の特徴を持つ。

- i 全体制御部と推論部にC言語を、利用者インターフェイスにCOBOLを用いることにより、Prologで表現されたプロトタイプに比べ処理速度と扱えるソースプログラムのサイズの点で改善される。
- ii 対処するエラーコードの種類を約2倍にする。この際に京都大学大型計算機センターで発生したエラーの頻度の高い50種と試用結果から得られたエラーコードを考慮する。
- iii 推論順序において過去の相談結果を反映するようにする。

```

---RESULT--- プログラム相談システム (コンパイルエラー
コマンド ==>                                     確認 ( Y
-----
JZK424I-S ISN=6 ISN : これはFORTRANの文とはみ
からの波及エラーだとおもわれます。
-----
    6      INTEGER    PPP,MMM(10)
>>  6      INTEGFR   I,SUM(20)
    7      INTEGER    ASDF,ARRAY(20),EXCEEDS6
    8      CHARACTER*1 KKJ
    9      CHARACTER*1 LLL
   10     DATA YYY/1.0/
   11     DATA JJJ/'A'/
   12     DATA I,ARRAY/10*0,1,2,3,4,5,6,7,8,9,0,1,2,
   13     DATA KKJ/1/
   14     J = 4
>>  15     SUM(1) = 1
   16     DO 10 I=2,20
   17         SUM(I) = SUM(I-1) + ASDF

```

図2-1 正しい相談結果を出力した例

```

---RESULT--- プログラム相談システム (コンパイルエラー
コマンド ==>                                     確認 ( Y
-----
これは文関数定義文でしょうか？
文関数定義文は実行文の前に書く必要があります。
移して下さい
-----
    9      DO 10,N=3,M,2
   10         DO 20,I=3,INT(SQRT(REAL(N))),2
   11             IF(N.EQ.N/I*I) GO TO 10
   12     20      CONTINUE
   13     P=P+1
>>  14     TBL(P)=N
   15     10     CONTINUE
   16     DO 30,N=1,P,10
   17         TBL=N+9
   18         IF(P.LT.TBL) TBL=P
   19         WRITE(6,300) TBL(N)
   20     300    FORMAT(1H0,10X,10I10)
   21     STOP

```

図2-2 見当がいの相談結果を出力した例

#### 謝 辞

本システムのプロトタイプの試用をしていただいた富士通電算機専門学院の指導員ならびにプロジェクト関係者各位に感謝致します。

#### 参考文献

- [1] 大西他：コンパイル時におけるプログラム相談の自動化、電子情報通信学会技術研究報告AI88-32 (1988)
- [2] 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)
- [3] 佐藤他：プログラム相談プロジェクトConsultの全体構想、情報処理学会第38回全国大会3N-7 (1989)
- [4] 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)
- [5] 佐藤他：実行時エラーのプログラム相談システム『Consult:R』情報処理学会第39回全国大会発表予定 (1989)