

初心者の作成したCOBOLプログラムの 誤りに関するデータベースとその利用

江木鶴子 岡村健史郎 長田一興

宇部短期大学 大島商船高専 近畿大学九州工学部

筆者らは、COBOL入門コースで学生が作成した誤りプログラムを対象に、初心者の起こすプログラミングの誤りに関して、①実行時に異常終了する誤りの頻度と原因、②ゴール/プラン分析法によるゴール別の誤り傾向、の2つの側面から今まで分析した。今回それらの分析結果に基づいて、プログラムの誤りカタログを作成し、誤りプログラムと共にそれらをデータベース化した。本データベースの特徴は、学生が実際に起こした誤りプログラムが格納されていることと、学生、教師、研究者からの利用を考慮して、誤りの指導と誤り訂正過程の情報も含めていることである。本論では、データベースの構築とその内容、データベースの一部を教育に利用した経験について述べる。

Construction of a Bug Catalogue Database and
its Use for COBOL Programming Education

Tsuruko EGI

Kenshirou OKAMURA

Kazuoki OSADA

Ube College
Dep. of Computer Science

Oshima National College
of Maritime Technology

Kinki University
Faculty of Engineering in Kyusyu

In this paper we describe a bug catalogue database stored with buggy programs and catalogues of errors generated by beginners. We constructed this catalogue on a base of two kinds of analysis of programming errors; analysis of frequency and factors of errors in those programs which were syntactically correct but terminated abnormally and goal/plan analysis of programmers' intention. We also describe some experiences of this database in order to assess its practical value in programming education.

1. はじめに

プログラミング教育における学習者の誤り情報（以後バグ情報と呼ぶ）は、教師、学習者、研究者それぞれにとって有用な情報である。にもかかわらず、教育の現場で発生する大量のデータを収集しにくいことや、大量のデータが収集できてもその中から精度の高い情報を抽出することは非常に煩雑な作業を要することなどから利用しやすいバグ情報として蓄積されたものが少ない。筆者らは、プログラミング教育の環境改善をめざす目的で、COBOLプログラミング教育で収集した学生プロトコルを調査し、そのうち2つの側面からの分析結果を報告した^{1)・2)}。現在これらの分析で得られた誤りの傾向や頻度を基準として誤りプログラムに関するデータベースを作成している。本論では、このデータベース（バグ・カタログデータベースと呼ぶ）の内容とその利用について述べる。

2. バグ・カタログデータベースの特徴

バグ情報を教師、学習者、教育システム作成者の立場からみると、次のような役割がある。

(1) 教師からみた役割

プログラミング教育をする教師の活動は、学生の起こす誤りから学生個人の理解程度を推測し、学生のプログラミングにおける過誤や誤認識を明確にして、それに対する指導を行い、さらにそれらの指導における効果を判断し、新たな指導を実践することである。このような活動を通して利用される教授知識は、現在のところ教師個人の知識としてしか蓄積されていない。したがって、学生プロトコルデータなどを分析した知識を広く利用できるような形の情報として蓄積することは、学習者がプログラミングを認知する過程を知る手掛りになるだけでなく、医者が診断する際に多くの症例を参考にするように、教師の指導を支援する情報源となる。

(2) 学習者からみた役割

プログラムは作成者の意図通りに即機能するわけではなく、プログラム完成までにデバッグは欠かせない。初心者はこれらの作業に使う知識がほ

とんどないうえに、計算機世界の知識も不足している。学習するという立場からすれば、これらの誤りをきっかけに誤りのメカニズムを理解していくことがプログラミング学習そのものであり、バグ情報は学習を促進する重要な情報である。

しかし、これらに関する情報が各プログラミング教育環境に学習者が利用できる形で整備されているとは言い難いのが現状である。既製のエラー表やマニュアルなどだけでは、初心者の場合誤り箇所がある程度判別できても、その原因メカニズム、つまりなぜこの箇所が誤りなのかを理解することは容易ではない。実習時の経験では、学生間で誤った原因メカニズムの情報を交換し、多くの学生がそれで納得していることがよくある。したがって、この重要なプログラミング学習過程において教育的に整理されたバグ情報を学習者に提供することは、教育効果をあげるために必要な教育的手段である。

(3) 知的C A I作成上の役割

知的C A Iは、それぞれの教育分野の知識がシステム構築の重要な要素となる。したがって人間の教師がほぼ無意識に保有し実践している知識や教授方法、学習者の理解度に関する判断基準や学習者のふるまいなどを明確にすることがシステム構築には欠かせない。プログラミング教育のための知的C A Iにとっては、このような情報のひとつが初心者の起こすバグ情報である。これらはシステム構築の基礎情報となるばかりでなく、作成されたシステムの評価をするための基礎データとしても重要な役割を果たす。

主に(1)の目的でプログラムの誤りを分析し、その結果を報告した例はいくつか存在するが、初心者が起こした誤りそのものを何らかの分析結果にもとづいて蓄積した報告は非常に少ない。すでに報告されている例としては、エール大学で作成されたPascalの「BUG CATALOGUE: I」、「BUG CATALOGUE: II III IV」³⁾がある。この情報は初心者が作成したプログラムの誤りを診断するシステムPROUST^{4)・5)・6)}作成の基礎情報として重要な役割を果たしており、主に研究者の利用を目的に作成さ

れたバグ情報である。

筆者らの作成したバグ・カタログデータベースは宇部短期大学のCOBOLプログラミング教育で学生が起こした誤りプログラムを基にして構築したものであり、上記3つの面からの利用を目的にしているのが特徴である。したがって後述するように、情報の内容として誤りの分類や説明だけでなく、誤りに対する対策や指導の情報も含まれている。また、誤りの訂正過程の情報も格納しており、誤りの時系列変化についての情報も格納している。

3. 誤り分析

3.1 誤り分析対象データ

誤り分析の対象となったプログラムは、COBOL入門コースで実施した12個の課題である。これらの課題は、印刷、グループ処理、照合、チェックなどプログラミングの初期に出題されることの多い典型的な問題である。分析対象として、実行時の誤り分析ではすべての課題を対象とし、ゴール/プラン分析では始めから3番目に提出させたグループ処理を伴う印刷プログラムを対象とした。分析データの収集方法は、学生がディバグのためにジョブリストを表示するたびにそのリストが特定のファイルに自動的に収集されるようにした。

誤りの分析調査については、プログラミング教育に直接役立つという意味で、まず実行時の誤りに関しての分析を実施し、その後ゴール/プラン分析法を使って論理的誤りに関する分析を行った。また、ゴール/プラン分析法による分析で顕著になった誤りがその後どのように訂正されるかの過程も調査した。筆者らが作成したバグ・カタログデータベースは、これらの分析に基づいた誤りプログラムとその誤りに関する情報をデータベース化したものである。以下にその分析方法とデータベースの内容を述べる。

3.2 実行時に異常終了する誤り分析

分析データとしては、全学生プロトコルから各学生が一つの課題を完成させるまでに実行したプログラムバージョンの中で実行時に異常終了した

最後のプログラムを対象とした。この際、異常終了した箇所の命令が異なる場合は、それぞれをサンプリングした。これによって433本のプログラムが抽出された。

これらのプログラムを対象に、誤り原因の内容を抽出すると共に、それらの誤りを8つのカテゴリに分類し、その傾向を分析した。さらにその誤りを課題別にも分類し、課題による誤り内容の傾向も調べた。表1は、誤り原因として取り出された内容をカテゴリ別にまとめた結果である¹⁾。

表1. 実行時に異常終了する誤り内容

カテゴリ	誤り原因の内容
入力	(1)入力力ファイルのレコードが定義が誤っている (2)入力力ファイルのの終了時の処理を誤っている (3)入力力ファイルのから入力した数値を誤っている (4)出力力ファイルのOPENされていない。
出力	(1)出力力ファイルのに出力して閉じられている。 (2)入力力ファイルのに出力して閉じられている。 (3)出力力ファイルのCLOSEを2回している。
初期値	(1)配列の領域が初期化されていない。 (2)変数の領域が初期化されていない。 (3)変数の領域が初期化されていない。 (4)出力力ファイルの領域が初期化されていない。
データ型	(1)文字数値項目の値を数値項目に入れている。 (2)文字数値項目の値を数値項目に入れている。 (3)文字数値項目の値を数値項目に入れている。
繰り返し	(1)繰り返しの条件が初期設定されていない。 (2)繰り返しの条件が初期設定されていない。 (3)繰り返しの条件が初期設定されていない。
命令	(1)PERFORM... a) THRU句の指定がない。 b) 実行範囲外にジャンプする。 c) 実行範囲外にジャンプする。 d) 実行範囲外にジャンプする。 (2)IF ... a) 命令の終了が不明瞭。 b) 命令の終了が不明瞭。 c) 命令の終了が不明瞭。 (3)未定義のサブルーチンプログラムをCALLしている。 (4)パラメータの設定をCALLしている。
流れ	(1)照合処理の手続きが正しいかどうかを確認していない。 (2)必要な処理がある。 (3)必要な処理がある。 (4)必要な処理がある。 (5)必要な処理がある。
環境	(1)ジョブ出力の制御が適切に行われていない。 (2)ジョブ出力の制御が適切に行われていない。 (3)ジョブ出力の制御が適切に行われていない。 (4)ジョブ出力の制御が適切に行われていない。

これらの誤りが実行過程の初期に起きると、特に学生の戸惑いが大きいことから、初期の課題6つに関して、表1の誤り原因を基準にしてその事例を誤りカタログとして作成した。現在、データベースに格納されている実行時に異常終了する誤りカタログ件数は44件である。

3. 3 ゴール/プラン分析法による誤り分析

分析では、まず50人の学生プロトコルの実行終了状態を学生毎に時系列にならべ、完成までの実行回数が極端に少ない学生と最終実行状態が正常終了していない学生を除いた35人の学生を対象として選択した。次に各学生が実行したプログラムのうち翻訳エラーのなくなった始めてのプログラム35本を分析データとして抽出し分析対象とした。

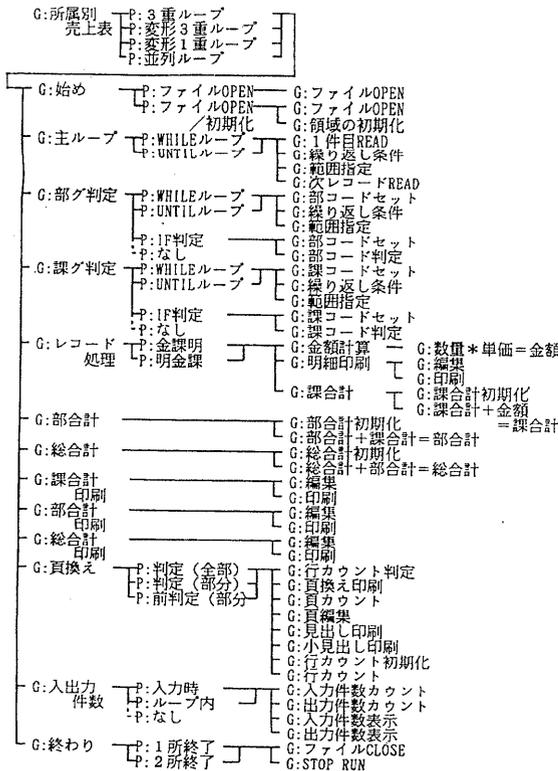


図1. 課題プログラムのゴール/プラン木

ゴール/プラン分析法は、課題の目標機能であるゴールとそれを実現する方法であるプランを階層的な木(ゴール/プラン木: goal/plan tree: GAP木)として表現し、これを基準にしてプログラムの誤りを分類し分析する方法である。

この分析で対象となった課題の内容は、「売上データ(所属、担当者、品名、数量、単価ほか)から所属別の売上一覧表を作成する」もので、これがトップゴールとなる。さらにこれを実現する際に「①数量と単価から金額を計算する、②ページの始めに見出し印刷し、1ページ55行印刷したらページ変えする、③所属(部、課)が変わるごとに金額の合計を印刷し最後に総合計を印刷する、④入力と出力の件数を数える」ことを指示しており、これがサブゴールを数え上げる際の条件となる。次に各ゴールの実現法(プラン)を学生プロトコルから抽出し、図1のGAP木を作成した。

このGAP木の葉ゴールを基準にしてカタログを作成した。作成されたカタログは、現在210件である。

3. 4 誤り訂正過程調査

翻訳エラーのなくなった始めてのプログラムにある誤りゴールが、その後どのように訂正されていくかを、図1の13個のサブゴール(G:始めからG:終わりのレベルのゴール)別に調査した。調査対象となったプログラムは355本である。さらにこの調査結果を視覚的にみるために、上記13個のサ

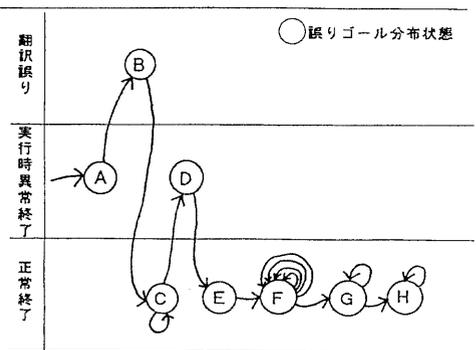


図2. 誤り訂正過程の状態遷移(例)

ブゴールのうちの誤りゴールの存在分布をひとつの状態とみて、学生別に図2のような状態遷移図を作成した。例えば図2のE状態は、G:課グ判定、G:頁換え、G:終わりに誤りがある状態で、F状態はG:課グ判定とG:終わりゴールにまだ誤りがあり、その後4回実行されているが改善されないままである。この4回の実行時に学生は、誤りゴールと関係のないG:頁換えとG:総合計印刷を3回に渡って訂正しているが1回は何も変更しないまま実行している。また、H状態はこの学生の最終状態であるが、まだ誤りゴールが存在している。

この状態遷移図では、プログラムの実行は行われているが誤りゴールの分布が変わらない場合、状態が推移しない。したがってこの状態を、学生がそのゴールの改善に集中しているかいないかにかかわらず、学生にとっての手づまり状態とみて比較的この停滞状態が大きい(図2ではF状態)と思われる箇所があるデータを選んだ。抽出された学生数は28人、プログラム総数は248本である。

4. バグ・カタログデータベース構築の概略

データベースの一次情報であるジョブ収集ファイルは、約50人の学生が作成した12個の課題に関する学生プロトコルである。したがって大量に蓄積されているうえ、内容がジョブリストそのものであるため非常に処理対象のデータとしては扱いにくい。そこでまず、①各プロトコル毎に学生ID、課題番号、実行終了状態、課金情報など23項目を抽出し定形化した情報としてジョブ集計テーブルを作成する⁷⁾。この情報もデータベース化されており、その後②このジョブ集計テーブルを参考にして分析対象プロトコルを抽出し、プログラムの誤りに関する分析を行う。③バグプログラムファイルには、この分析結果を基準にしてジョブ収集ファイルから選択されたプログラムが蓄積される。一方、④それぞれの分析結果を筆者らがカタログとしてコーディングしデータを作成する。⑤それを端末から入力し、バグカタログテーブル、バグ訂正過程テーブルを作成する。

このうち①、③、⑤はデータ入力することで自

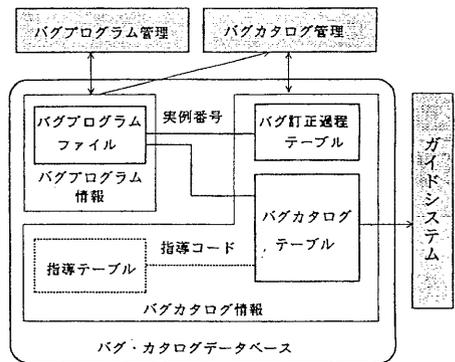


図3. バグ・カタログデータベースシステムの構成

動生成されるが、②の分析と④は現在のところ手作業である。

5. バグ・カタログデータベースの構造

5. 1 データベース全体の構造

バグ・カタログデータベースは、大別してバグカタログ情報とバグプログラム情報の2つに分けられる。バグカタログ情報には、バグカタログテーブルとバグ訂正過程テーブル、指導テーブルがあり、それらはリレーショナル型のデータベース管理システム上の表としてそれぞれ格納されている。バグプログラムファイルは通常の順編成ファイルとして存在している。これらの関係の概略を図3に示す。

バグカタログテーブルは、ひとつの誤りプログラムに関する情報が一組のデータとして構成されており、バグ訂正過程テーブルは一連の訂正過程の1段階のプログラムに関する情報が一組のデータとして構成されている。一連の訂正過程はカタログ番号と連番で順序化した複数の組で表現される。両テーブルとも実例番号をポインタとして、バグプログラムファイルの1つの誤りプログラムリストと関係づけられている。

5. 2 バグカタログテーブルの構造

バグカタログテーブルは、表2に示すようにカタログ番号、課題番号、誤り分析結果の情報(カテゴリ、タイプ)、その詳細、誤り現象、誤り説

明、対策、指導コードと指導内容、事例番号など13項目が含まれる。

カテゴリは、プログラムの誤りを表1に示す8種類に分類したものである。データベースには、この順に1から8のコードで格納されている。

カテゴリ詳細は、カテゴリだけでは不足な説明をするための項目である。ゴール/プラン分析から作成されたカタログには、例えば誤りの箇所やプランが、'P:3重ループ/G:課グ判定/P:whileループ/G:繰り返し条件(変形)'のようにGAP木の表現で記述されている。

タイプは、誤りプログラムを実行した際の現象を、「実行しない」、「中断する」、「結果が常に不良」、「結果が時々不良」、「不良プラン」の5種類に分類し、この順に1から5のコードで格納されている。誤りのタイプだけでなく、より詳細な説明が必要な場合には、タイプ詳細に記述する。例えば「中断する」の場合、どのような命令で中断したかを記録しておく。

表2. バグカタログテーブルの構造

項目	属性	内容
カタログ番号	文字 5桁	バグを特定する番号。登録順につけられた5桁の連番。
課題名	文字 8桁	誤り実例として格納されているプログラム課題名。
カテゴリ	文字 1桁	プログラムの誤りを8つのカテゴリに分類し、それを1から8までのコードで格納。
カテゴリ詳細	日本語50字	プログラムの誤り場所やプランなどカテゴリだけでは不十分な点に関する説明。
タイプ	文字 1桁	誤りプログラムを実行した際の現象を5つに分類し、それを1から5のコードで格納。
タイプ詳細	文字 10桁	誤りのタイプにより必要となる説明
現象説明	日本語50字	プログラム作成者の立場からみた誤り現象。
誤り説明	日本語25字	誤りの原因を実例プログラムにそって形で説明。
対策	日本語25字	誤りを取り除くための方法を実例プログラムにそって説明。
指導コード	文字 10桁	指導に関する説明のコード番号。
指導	日本語25字	このような誤りを起こす学生に欠けていると思われるプログラミングに関する指導内容。
時系列プロトコル数	数値	誤り訂正過程データの有無をプロトコル数で格納。
事例番号	文字 8桁	バグプログラムファイルに格納されている実例プログラムへのポイント

現象説明は、プログラム作成者の立場からみた誤りの現象を簡単な文章(50字以内)で説明したものである。例えば、「結果が出力されない」、「見出しと1行目だけを印刷し中断した」、「担当者一覧表とエラー表が途中で印刷されている」などである。

誤り説明は、実例プログラムにそってどのような理由で誤りが起こっているかを文章(125字以内)で説明したものである。つまり、この課題を作成している学生がこのような誤りを起こした際に、学生のプログラムを前にして教師が誤りの原因を説明するような内容を想定している。

対策は実例プログラムにそって誤りの訂正方法を具体的に記述したものである。

指導の項目には、実例プログラムにあるような誤りを起こす学生に欠けていると思われる知識を簡単に説明する。これは必ずしもプログラムの誤りと一対一に対応しておらず、ひとつの欠けている知識のために異なった種類の誤りを起こすことが多い。現在は指導内容が同じでも誤りの種類が異なれば、各誤りごとに指導情報を格納している。

表3. バグ訂正過程テーブルの構造

項目	属性	内容	
プロトコル番号	文字15桁	ユーザIDとプロトコル番号を組み合わせた番号。	
課題名	文字 8桁	誤り実例として格納されているプログラムの課題名。	
終了状態	文字 1桁	プログラム終了状態(翻訳エラー、中断、正常終了)1から3のコードで格納。	
出力推移	文字 1桁	プログラムが終了した時の出力結果の状態をAからZのコードで格納。	
実行日時	文字	そのプログラムが実行された日時。	
トッププラン	文字15桁	トッププランの名称。	
ゴール数の繰り返し	ゴール名	文字10桁	ゴールの名称。
	誤りの有無	文字 1桁	このゴールに誤りが発見されたか否かを1か0で格納。
	訂正の有無	文字 1桁	このゴールに訂正がされたか否かを1か0で格納。
バグゴール数	数値 3桁	このプログラムバージョンで発見されたバグゴールの数。	
訂正ゴール数	数値 3桁	このプログラムバージョンに訂正を加えたゴールの数。	
カタログ番号	文字 5桁	バグカタログテーブルへのポイント	
事例番号	文字 8桁	バグプログラムファイルへのポイント	

我々はこの指導内容がある程度蓄積された時点で、その内容を分類し指導テーブルに分割するつもりである。その際は分類された内容をコード化し、そのコードを指導コードに格納する予定である。現在はバグカタログ情報を登録するたびにこの指導情報も作成し、これによって指導情報を収集蓄積することになっている。

5. 3 バグ訂正過程テーブルの構造

先に述べたようにバグ訂正過程テーブルでは、訂正の一段階のひとつのプログラムがデータの単位として構成されている。それらの内容は表3に示すように、プロトコル番号、課題番号、終了状態、出力形態、トッププラン名、サブゴール別にサブゴール名、誤りと訂正の有無、バグゴール数、訂正ゴール数の13項目の情報から構成されている。このうち、ゴール別誤りと訂正の有無項目には、データ部とジョブ制御文での誤りと訂正の有無も格納される。

5. 4 バグプログラムファイルの内容

誤りの実例プログラムは、学生が課題作成時に実行したプログラムリストから抽出されたものもとになっている。通常このリストには、課金情報やジョブ制御文、翻訳状態、実行完了状態、実行結果などが含まれる。データベースに格納する実例プログラムは、これらのリストからデバッグに不必要と思われる情報、例えば課金情報や翻訳リストのページ見出し行などを除き、さらに誤りを引き起こしている行や誤りに関係がある行にコメントを付加したものである。リストの1行が1レコードとして構成されている。

6. バグ・カタログデータベースの管理

図3に示すように、バグ・カタログデータベースシステムには、バグカタログ情報とバグプログラム情報の管理に必要なツールが備わっている。

バグプログラム情報の管理は、主として一次情報であるジョブ収集ファイルからバグプログラムファイルを作成することである。これらの処理は

通常次の3段階で行われる。

①4. で述べたジョブ集計テーブルの情報を参考にしながら、ジョブ収集ファイルの部分集合であるファイル（スタックファイルと称する）を作成する。②通常はこのスタックファイルからさらに実例プログラムの候補プログラムを1本だけファイル（ピックファイルと称する）を抽出しこれを加工する。③ピックファイルの内容をバグプログラムファイルに格納する。

以上の処理に対して、次の7種類の機能が用意されている。

- a) ジョブ収集ファイルからユーザIDと課題番号を検索キーとして対象のプログラムを抽出
 - b) あるいは、リスト番号を検索キーとして抽出
 - c) スタックファイルからジョブリストの抽出
 - d) ピックアップファイルの編集
 - e) バグプログラムファイルの更新
 - f) バグプログラムファイルのプログラム抽出
 - g) バグプログラムファイルのプログラム抹消
- バグプログラム情報の管理はこれらのツールを組み合わせて行う。

バグカタログ情報の管理ツールとしては、各テーブルへのデータの a) 登録、b) 修正、c) 検索の他に d) インデックスリストの印刷、e) バグカタログリストの作成の5種類がある。このうちインデックスリストの印刷では、課題、タイプ、カテゴリ別にインデックスリストが作成できる。また、バグカタログリストの作成では、バグカタログの内容と対応する実例プログラムをバグ番号順に印刷することができる。

7. バグ・カタログデータベース利用の試み

バグ・カタログデータベースのうち、実行時に異常終了する誤りに関しての情報を教育へ直接利用するため、簡単な誤りのガイドシステムを作成した。冊子としてのバグカタログリストと共にこのガイドシステムを1991年前期に教育現場で試用したので、以下にその経験を述べる。

誤りガイドシステムは、課題名と異常終了した箇所の命令を端末から会話的に入力することによ

って、その誤りの原因と考えられる情報をデータベースから検索し、その内容を段階的に表示するという簡単なものである。

実習時に実行時の誤りに関して相談に来た学生に対して、誤りガイドシステムを使って誤りの原因としてデータベースに格納されている説明を検索し、学生にそれらと同じ誤りを起こしていないかを調べるように促す。これだけで誤りの訂正の仕方が分からない学生には、その対策や指導情報も見せる。これでもまだ理解できない場合は、冊子のバグカタログリストにある実例プログラムを参照しながら教師が説明をする。

相談数は1回の実習で2～3件程度で、予想したように始めの3つの課題に多かった。その中で、表1にある誤りであっても検索されない例が4件あった。いずれも別の課題ではその誤りカタログがあるが、検索で指定した課題の誤りとしてはデータが格納されていないことが原因であった。また、表示される文章だけでは、やはり学生にはかなり理解が難しいことも明らかになった。したがって、このシステムは学生が直接使用するより、むしろ学生のプログラム相談や指導をする教師の道具として利用の方が効果的である。

この誤りガイドシステムの他に、実習時には冊子体のバグカタログリストを教師が指導するための道具として備えていたが、特に後半の課題の方で、このリストを直接参照している学生の姿が何人か見られた。この効果についての正確な把握はできていないが、時間をかけて考えるようなディバグ時には、このような冊子の有効性は高いのかもしれない。

8. 考察

バグ・カタログデータベースの構築とその利用としての誤りガイドシステムについて、現状での考察を以下に述べる。

1) カタログデータベースは、実際に学生が起こした誤りを分析した結果が蓄積されているため、教育で使用するバグ情報のデータとしては質がかなり高い。しかし、誤りガイドシステムの使用で

述べたように量的にはまだ不足している。

2) データベースの量の問題を克服するためには、構築のツールを改善する必要がある。本論で報告したように、現在は収集されたプログラムを後で構築するという方法をとっているが、一応の分析を終了した後は、実習中に発見した新しい誤りプログラムをすぐその場で抽出し構築できる方が良い。また、誤り訂正過程調査などは調査そのものを援助するツールがないと研究に必要なデータ収集をするには限界がある。

3) バグ・カタログデータベースを直接教育へ利用した経験では、教師のための道具としてはかなり役立つことが分かった。しかし学生が利用する道具としては、予想していたことであるがもっとインターフェイスを高度に改善する必要がある。そのためには、学生がプログラムの誤りを表出されている情報のうち何を手がかりにどのように認知しているかなどを十分に分析する必要がある。誤り訂正過程調査から得られたデータは、これらの分析をするための有効な情報源となることが予想される。

参考文献

- 1) 江木、岡村、長田：COBOLプログラミングにおける初心者の誤り傾向、C A I 学会誌、VOL. 6, NO. 3 (1989)
- 2) 江木、岡村、長田：ゴール/プラン分析法による初心者の作成したCOBOLプログラムの誤り分析、C A I 学会誌、VOL. 7, NO. 2 (1990)
- 3) Spohrer, J. C., etc: BUG CATALOGUE: II, III, IV, Tech. Rep. 386, Dept. of Computer Science Yale Univ. (1985)
- 4) Johnson, W. L., Soloway, E.: PROUST: An automatic debugger for Pascal programs, BYTE, VOL. 10, NO. 4 (1985)
- 5) Johnson, W. E., Soloway, E.: PROUST: Knowledge-Based Program Understanding, IEEE, VOL. SE-11, NO. 3 (1985)
- 6) Johnson, W. L.: Intention-Based Diagnosis of Novice Programming Errors, Morgan Kaufmann Publishers Inc., (1986)
- 7) 江木、長田：COBOLプログラミング教育のための教材管理システム、宇都短期大学学術報告、NO. 24 (1987)