4G-8　実行時メッセージに基づくデバッグモデルとその適用例

高田司郎　米山寛二　野田昭司　鳥居宏次
㈱ＣＳＫ総合研究所　コンピューターサービス㈱　大阪大学基礎工学部

1．はじめに

近年、ソフトウェア工学に知識工学を取り込んだソフトウェア開発支援システムが開発されている。要求仕様の入力から、開発に必要な知識を利用したプログラムの半自動合成システムである。ところがこれらシステムを実用化する為には莫大な知識の投入を必要とし、一朝一夕に解決できる問題ではない。そこで筆者らは、単体テストフェーズに焦点を絞り経験の浅いプログラマーを対象としたデバッグ支援エキスパートシステムＩＴＴＢ（an Intelligent Tool for Taking-away Bugs）を開発、実用化を目指している。

ＩＴＴＢにおけるデバッグ支援とは、ソフトウェアの故障診断と見做し、現象として「実行時メッセージ」を入力し、故障診断として「対策案」を出力するものとする。そこで、実行時メッセージとなる条件コードがよく整備されている汎用プログラミング言語ＰＬ／Ｉを選択した。本稿では、システム構築に当り、実行時メッセージに基づくデバッグモデルを想定し、そのデバッグ知識を記述した文書（以下「直伝の書」と呼ぶ）を作成し、ソフトウェア開発現場で適用した結果、その有効性を確認したので報告する。

2．実行時メッセージに基づくデバッグモデル

通常、ＰＬ／Ｉプログラマーは、プログラムが異常終了した場合、実行時メッセージとして出力される条件コード（8097 = データ例外など）をトリガーとして、プログラムソース、実行結果及びダンプ、マニュアル等を参照し、過去の経験知識に基づいてプログラムのデバッグを行なっている。ＩＴＴＢにおいては、プログラムの異常終了時（つまり今回は要求仕様上のバグは除外している）に限定して、「実行時メッセージ」を入力とし「対策案」を出力とする。よって当ＩＴＴＢ利用者は、異常終了した条件コードを入力する。その後ＩＴＴＢは、蓄積した経験則に基づいて、質問応答を行ない原因候補を絞り込み、その原因に対する対策案が優先順序付きで出力される。ユーザーは、その対策案に基づいて、プログラム、データ又はＪＣＬなどを修正し再実行を試み、その異常終了が解消されるまで対策案を繰り返す。但し、修正内容によっては他の異常終了を誘発することもあり、この場合は、入れ子処理となる。（実際の修正はユーザーが行なう為、ユーザーが理解できる対策案を出力する必要がある。例えば、プログラム例など）

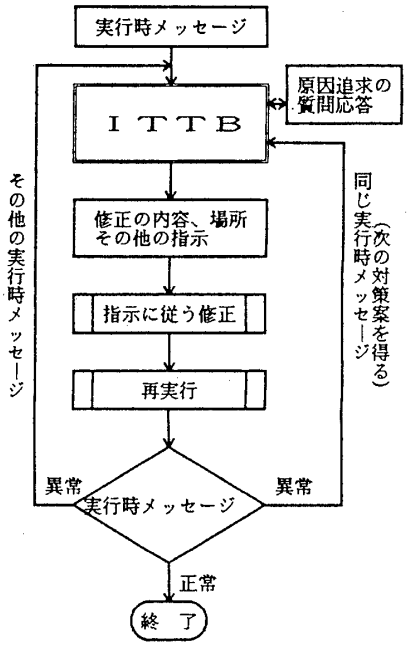

図１　デバッグモデル

3．システムの近似的手順

M_i ：条件コードｉの実行時メッセージ

P_{ti}：実行時メッセージM_i に対する原因の候補

$q_{ti,k}$：原因P_{ti}に対するｋ番目の対策案の候補

と定義すると、システムの近似的手順は次の様になる。

（１）経験則より原因の候補をP_{ti}に絞る。

（２）原因P_{ti}に対する対策案の候補から優先順位の高い順に$q_{ti,k}$を採用する。

（３）$q_{ti,k}$を全て採用してもメッセージM_i が消えないならば別の原因の候補P_{ti}を選び（２）をくり返す。

Debugging Model on the basis of Execution Messages and its Application
Shirou TAKATA , Kanji YONEYAMA , Shouji NODA , Kouji TORII
CSK Research Institute , COMPUTER SERVICES Corp., OSAKA Univ.

（4）メッセージM_iが消えたが新たなメッセージM_i'が発生したならば原因の候補P_{ij}を選び（2）をくり返す。

4．「直伝の書」の作成

上記の近似的手順を検証する為にＰＬ／Ｉプログラムテスト経験者８名に対して、条件コードの発生頻度を回答させ、とりまとめた結果を図２に示す。この結果より発生する条件コードはほぼ類似しており、プログラマーがよく犯す誤りの傾向を示唆している。この中から発生頻度の高い37項目について「現象→原因→対策」の知識を集約した一覧表を作成した。これはデバッグ活動の経験的手法として再利用可能なので「直伝の書」と呼ぶ。データ例外を例として、「直伝の書」を図３に示す。この中で、対策は発生頻度の高い順に記述した。又知識はデバッグ知識をチューターとしてプログラマーに提示し、理解させる為に記述した。さらに直伝の書作成の副次的効果として、利用の手引を添付して２週間に渡りソフトウェア開発現場のテスト担当者34名に配布し、活用してみたが以下の点で十分利用できる見通しを得た。

（1）エラーの発生原因が図２に示した条件コード内でほぼ閉じている。

（2）初心者にとって原因列挙の指針となり上級者とのレビューを可能とした。

（3）上級者にとって原因の全体像の再確認ができると共に初心者に対する同品質の指示及びその時間の短縮が計れた。

5．おわりに

ＰＬ／Ｉにおける実行時メッセージに基づくデバッグ支援システム構築の為に、デバッグモデルとデバッグ知識の整理について述べた。現在、この結果をもとにエラーを生じるプログラム例、及び正常なプログラム例を作成中である。今後、エキスパートシステム開発支援ツールを用いてプロトタイプをインプリメントする予定である。

6．参考文献

(1)Johnson,L. and Keravnou,E.T.,Expert Systems Technology,Abacus Press, 1985.

(2)OS および DOS PL/I 言語解説書，日本ＩＢＭ.

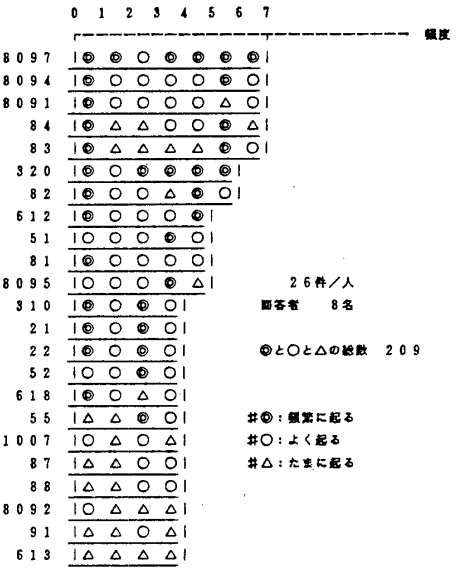

図２　条件コードアンケート　調査結果

条件コード	現　象	原　因	対　策	知　識	知識の種類	マニュアル
8097 (IBM5371)	正しい形式になっていない FIXED DECデータの処理をしようとした。	1. エリアの種類別原因 (1)ワークエリア －初期値化していない (INITもしくは代入) －他の属性で再定義して使用 (2)INPUT ファイル －レコード内容の不良 －READ処理のエラー ・読み込みのタイミング －レイアウトのずれ (3)サブルーチンのパラメータ －パラメータ指定ミス 2. プログラムでエリアをこわした。 －SUBSTR －SUBSCRIPT　の 添字オーバー	1-1 －初期値化する。 －再定義での誤った使用を訂正する。 1-2 －レコード内容を修正する。 －READ処理を正しく書き直す －レコード内容とレイアウトの定義を一致させる。 1-3 －パラメータの受け渡しを訂正する。 2. －STRINGRANGE －SUBSCRIPTRANGE を禁止状態にする。 ※　ON条件でひっかけるエリア破壊の箇所を見つけ訂正する。	第1段階 －パック十進数の意味，十進演算とは何かが分からない ※　初期値化の必要分かる　性を知る。 第2段階 －十進演算は分かるが演算の前提となる知識を持っていない。 ※　パック十進数がはい獲得　っていない原因を追跡できる。 第3段階 －十進演算を理解し演算の前提を満たしていることも確認できるが、エリア破壊という状況を知らない。 ※　SUBSTRの知る。　配列扱い		

図３　「直伝の書」　（データ例外）