



Title	実行時エラーメッセージに基づくデバッグ用エキスパートシステムについて
Author(s)	野村, 研仁; 井上, 克郎; 鳥居, 宏次 他
Citation	全国大会講演論文集 第35回昭和62年後期. 1987, 2, p. 1493-1494
Version Type	VoR
URL	<a href="https://hdl.handle.net/11094/50400">https://hdl.handle.net/11094/50400</a>
rights	ここに掲載した著作物の利用に関する注意 本著作物の著作権は情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。
Note	

*The University of Osaka Institutional Knowledge Archive : OUKA*

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

## 実行時エラーメッセージに基づく デバッグ用エキスパートシステムについて

4L-2

野村研仁\*  
木村陽一\*\*

井上克郎\*

鳥居宏次\*  
米山寛二\*\*\*

\*大阪大学基礎工学部

\*\*株 C S K

\*\*\*株 C S K 総合研究所

1. はじめに

プログラムの実行時エラーの原因を効率よく発見するためには、デバッグに関する多くの経験的知識が必要である。熟練したプログラマーは、処理系が発生するエラーメッセージを手がかりにしてエラー原因の見当をつけ、ソースプログラム・実行結果等を詳しく解析することにより、見当をつけた原因の候補の検証をおこない、エラー原因を特定するという方法でデバッグをおこなうことが多い。

初心者プログラマーが効率よく経験的知識を獲得し、その知識を効果的に用いることができるようになるとには、熟練者の適切な支援・指導を受けることが不可欠である。しかし、熟練者が初心者に常に適切な支援・指導をおこなうことができる環境を作ることは容易ではない。そこで、熟練プログラマーのデバッグに関する経験的知識を用いたエキスパートシステムがあれば、初心者は必要な時にシステムと対話することにより、効率よくデバッグ作業をおこなうことができると期待される。

ソフトウェアの開発において、広く使用されているプログラミング言語の一つにPL/Iがある。PL/Iは機能が非常に豊富な言語である。初心者プログラマーは、これらの豊富な機能を正確に理解せずに使用して誤りを犯すことがある。機能の理解不足によって発生した誤りの原因を追求し、その誤りを取り除くことは、初心者でなくともきわめて困難な作業である。

したがって我々は、初心者のデバッグ作業を支援するため、プログラミング言語PL/Iを対象とした実行時エラー原因診断システムの試作をおこなった。

本稿ではデバッグ用いる経験的知識と本システムの概要について述べる。

2. 経験的知識を用いたエラー原因の推測

言語解説書等は、エラーメッセージの説明やエラーの直接原因について記述しているものがほとんどで、これらからプログラムのどこを修正すれば良いかを知るのは困難であると考えられる。プログラムを修正するために、プログラマーがどのような‘本質的’な誤りを犯したためにエラーが発生したかを知ることが重要である。本質的な誤りとは、その誤りをプログラマーが認識することによって初めて修正ができるようなものをいう。以下ではこのような本質的な誤りをエラー原因と呼ぶことにする。熟練したプログラマーは、エラーの原因に関する知識を長年のデバッグ作業を通して獲得している。たとえば、データ例外というエラーメッセージが発生した場合、熟練したプログラマーは、初期化していない変数を参照しようとしたことが原因ではないかと見当をつける。これは、初期化していない変数を参照しようとしたためにデータ例外エラーを起こす事例が非常に多いことを、経験的に知っているからである。これに対し、マ

ニュアル等から得られる知識では、固定小数点10進数データの内部表現は、最下位の4ビットが符号をあらわす特殊なビットパターンにならなければならぬことしかわからぬ。これでは、この部分が不正な値になっているためにエラーが発生したということしかわからぬ。

したがって、エラーメッセージとエラーの原因を結び付ける経験的な知識を熟練プログラマーから獲得し、整理して有効に活用することが重要であると考えられる。この考えに基づいて、デバッグ支援システム構築のための知識の整理がすでにある程度おこなわれている<sup>[4]</sup>。しかしこれはエラーの原因等に関する記述が不十分なので不足している知識の補充・知識の詳細化をおこない、エキスパートシステムで用いることができるようプロダクションルールの形でルールベース化した。

3. データ例外エラーの原因

PL/I処理系は多様な実行時エラーメッセージを発生するが、今回はその中からデータ例外(data exception)を選び、その原因を診断するシステムを作成した。データ例外とは、正しい形式にならない固定小数点10進数データを処理しようとした時に発生するエラーである<sup>[5]</sup>。このエラーは、計算機内部のデータ表現やデータ操作を正しく詳細に理解していくと、エラー原因を発見することが困難なエラーであるが、デバッグに関する経験的知識を用いることによりかなり正確・迅速に原因を発見することができる。また、熟練したプログラマーに対する調査の結果、頻繁に発生するエラーであることがわかっている。したがって今回、試作のためのエラー例としてデータ例外を取り上げるのが適当であると考えた。我々は、熟練したプログラマーに対する調査や、PL/I言語解説書の検討をおこなった結果、データ例外の原因を8項目に分類した。各原因を以下に示す。

- 1) 変数を初期化していない。
- 2) 配列変数の添字値が範囲を逸脱した。
- 3) 記憶領域を共有している変数の型や大きさが異なっている。
- 4) 変数の記憶領域が破壊された。
- 5) 関数の実引数と仮引数の型が異なっている。
- 6) 変数が構造体のメンバーで、その変数が属する構造体に異なる型を持つ値を代入した。
- 7) 構造体でない変数に構造体を代入した。
- 8) 変数が入力データの受取用で、入力されたデータが予期しない値であった。

4. エラー原因診断エキスパートシステム

実行時エラー原因診断エキスパートシステムは、PL/Iプログラムが実行中に異常終了した時に、エラーメッセージ、ソースプログラム等から得られる情報をもとにエ

ラー原因の診断をおこなう。システムは、データベース初期化部・知識ベース・推論部・出入力インターフェース部によって構成されており、プロダクションルールによるルール表現・フレーム形式による知識ベース構成に基づいて作成されている。  
(図1)。なお本システムが対象としたPL/I言語は、IBMのOS PL/Iである。

データベース初期化部は、データベース内の推論の中間結果・最終結果に関する情報を格納する場所の初期化をおこなった後、知識ベース内のデータベースにエラーメッセージ・PL/Iソースプログラムの解析結果・実行時情報を書き込む。そして、推論部に対して推論開始メッセージを送る。

知識ベースは、データベースとルールベースによって構成される。データベースは、フレーム表現を用いて作成されている。各フレームには、必要な属性を格納するスロットが用意されている。フレームは、その性質にしたがって階層的に構成されており、上位の概念は下位に継承される。現在、データベースには、プログラム全体に関する情報を持つフレーム・各変数に関する情報を持つフレーム・推論の最終結果を格納するフレームが用意されている。

推論規則は、仮説生成規則と仮説検証規則に大別される。仮説生成規則は、LISP言語の関数の形で記述されている。仮説検証規則は、プロダクションルールの形で記述されており、後ろ向き推論を用いて検証がおこなわれる。仮説生成規則は、データ例外に関するエラー原因の中から可能性のある候補を生成する規則である。すべての候補について、一つ一つ後ろ向き推論を用いて検証していくと効率が悪いので、まず仮説生成規則を用いて発生する可能性のない原因候補を除去し、検証すべき候補を絞り込む。仮説検証規則は、仮説生成規則を用いて選び出された候補の中から、各変数に関する情報等を用いて候補の検証をおこない、エラー原因を特定するための規則である。これらの規則の構築にあたっては、規則に階層性を持たせ、規則間の関係が容易に理解でき、追加・削除・変更などをしやすいように注意した。

本システムでは、ユーザーのシステムに対する命令や質問は、基本的には、イメージパネル上のメニュー・スイッチをマウスを用いて選択することによりおこなう。また、推論結果の表示もイメージパネル上におこなわれる。イメージパネルとは、ユーザーインターフェースを向上させるため、システムの状態や各種メニューなどをディスプレイ上に图形的にあらわしたものである。具体的には、推論開始スイッチ・推論過程説明起動メニュー・推論結果(エラー原因)の表示・システムの動作状況の表示などに用いられる。また、推論過程の説明は、専用のウィンドウ上でおこなわれる。

### 5. あとがき

本システムは、ワークステーション XEROX 1108 上のエキスパートシステム開発支援ツール KEE の下で稼動している。ルール数は、仮説生成のためのルールが 7、仮説検証のためのルールが 16 である。各種手続きは、Interlisp-D を用いて記述されている。本システムは、データ例外エラーに関する 8 項目 14 種類の原因診断をおこなうことができる。本システムの作成のため、エラー原因の調査・検討・整理に 3 ヶ月、システムのプログラミングに 1 ヶ月を要した。

現在、診断に利用する情報をユーザーが入力しているが、たとえば初期化文の認識などはある程度自動化が可

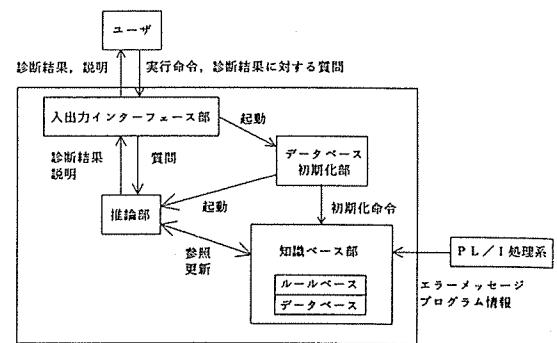


図1 エラー原因診断エキスパートシステムの構成

能なので、今後このような情報の自動獲得をおこないたい。自動化の一つの手がかりとしてPL/Iコンパイラの情報や実行結果のダンプリストの利用があげられる。これに伴って本システムをPL/Iが作動する環境上に移植する方が都合がよくなると考えられる。また診断できるエラーの種類を増やしていくことも実用性を高める上で重要である。発生頻度の高い三十数項目のエラーについては、ある程度デバッグに関する知識の整理がおこなわれているので、ルールベース化するために、知識の詳細化・不足している知識の追加・知識の内容に基づいた整理等をおこなうことにより、これらの知識をエキスパートシステムで利用できると考えられる。

**謝辞** 本研究にあたり、多數の有益な御助言御協力をいただきました(株)CSKの西良厚正氏に感謝致します。

### 参考文献

- [1] IntelliCorp: "KEE Software Development System User's Manual" (1985-07).
- [2] XEROX: "Interlisp-D Reference Manual" (1985-10).
- [3] 伊藤博樹: "PL/I プログラムにおけるデバッグ支援システムの作成", 大阪大学基礎工学部情報工学科特別研究報告 (1986-03).
- [4] 高田, 米山, 野田, 鳥居: "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", 情報処理学会第33回全国大会, pp. 751-752 (1986-10).
- [5] 仲田恭典: "PL/I プログラム実行時エラーの原因診断エキスパートシステムの試作", 大阪大学基礎工学部情報工学科特別研究報告 (1987-03).
- [6] 日本IBM: "OS および DOS PL/I 言語解説書" (1985).
- [7] 西良, 木村, 米山, 鳥居: "エキスパートシステム構築支援ツール CRI-EXPERT(3) -応用事例(1)-", 情報処理学会第35回全国大会, 1N-6.
- [8] 野村, 仲田, 井上, 鳥居, 木村, 米山: "実行時エラーの原因診断エキスパートシステムの試作", 情報処理学会ソフトウェア工学研究会, 87-SW-54-3 (1987-06).