アシスト機能を備えたプログラム開発環境

1S-4

°山本 壮一, 笠原 道治, 黒木 伸一, 渡邊 豊

(富士通株式会社)

1. はじめに

我々は, プログラマがプログラム開発の一連の作業を各ツールを意識することなく, 自然な操作ビューで進められる環境 (「Cプログラマブラウザ環境」[1,2]と呼ぶ) を作成した (図1参照). これは社内及び社外のプログラマに使用されており, 高い評価を得ている.

しかし, この環境下においても翻訳結合でエラーが発生した時に対処が分からず困っているプログラマが多い. 我々はこの状況を何とか解決できないかと考えた.

そこで, 対処が分からない時にプログラマがどうしているかを調査したところ以下のことが分かった.

① マニュアルを見たり上級者に聞いたりして解決している.

② 対処法はそのプログラマのスキルや経験によって違っている.

このことから, マニュアルに記載されている内容や, 上級者が持っているノウハウをプログラマに知らせる機能を提供すれば, プログラム開発の効率が上がると考えた.

我々は, そのプロトタイプを作成し, 試行したのでその結果を報告する.

2. アシスト機能

翻訳結合のエラーの中にはエラーメッセージを見ただけで単純に対処法が分かるものと, 簡単には対処法が分からないものがある. 対処法が分かりにくい理由として

① 1つのエラーに対して幾つかの原因が考えられる.

② エラーの出た行にエラーがあるわけではない.

の2つが考えられる. この場合の対処法はマニュアルには書きにくいノウハウのようなものが多い.

これらのノウハウをプログラマとの対話形式で表示できるものがあれば, 初心者のプログラマでも上級者のように対応できると考えた. このような機能を我々はアシスト機能と呼ぶことにした.

エラーとその対処法には様々なパターンがあるが, 分類してみると, 図2に示す3つのパターンになった.

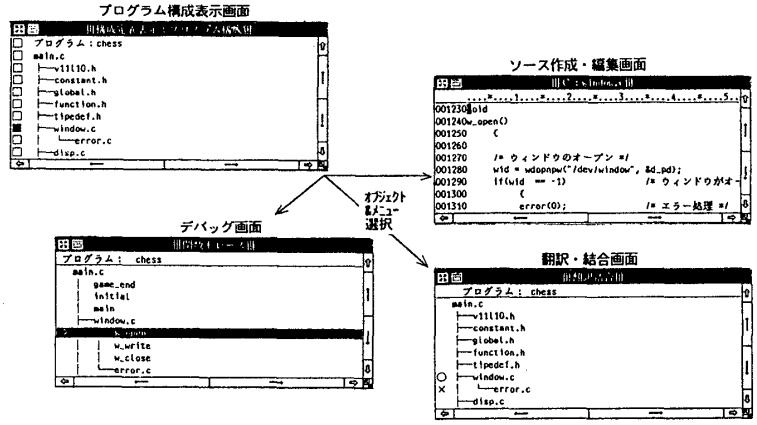


図1. Cプログラマブラウザの操作画面

Programming Environment Which Assists Programmers
Souichi Yamamoto, Michiharu Kasahara,
Shinichi Kurogi, Yutaka Watanabe
FUJITSU, Ltd.

その最も単純なパターンとして図2(a)に示すものが考えられる．しかし，必ずしも(a)のように単純に一対一に対応するだけでなく，(b)のように複数の対処法から1つの対処法を選択する場合もある．また(c)のように複数のエラーから対処法が決まる場合もある．

これら3つのパターンから，以下に示す情報の構造を考えた（図3参照）．

① 各情報は識別子を持つ．これを我々は情報ＩＤと呼ぶ．

② 各情報は，対処法と関連ある対処法の情報ＩＤ（0個以上）から構成される．

③ 各情報はネットワーク構造をとることができる．

これをアシスト情報と呼ぶことにした．

3．効果

プロトタイプを作って社内で試行した結果，このアシスト機能によって

① 上級者のノウハウを初心者が簡単に利用できる．

② 上級者にとっても特殊なエラーのときは効果がある．

③ ノウハウを蓄積し，共有することができる．

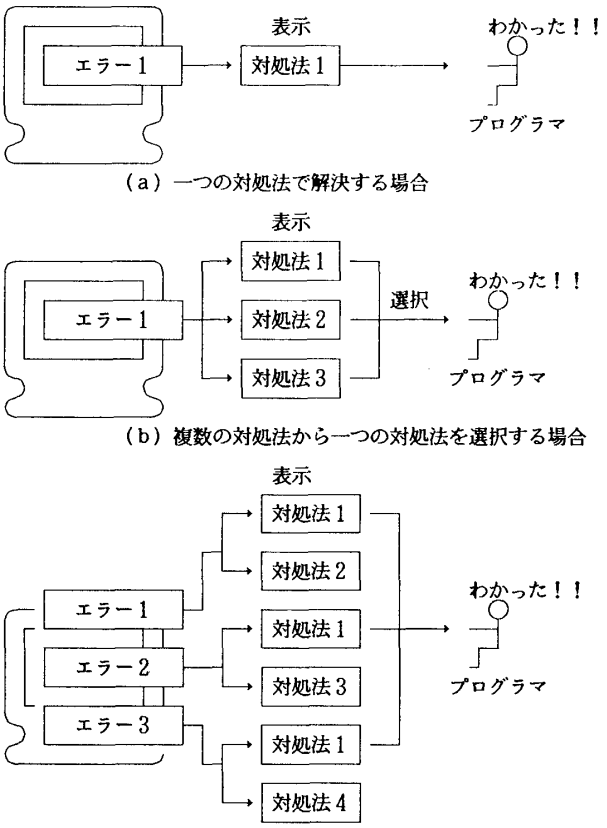
④ マニュアルを見なくてもエラーの意味・対処法が分かる．

などの我々のねらった効果が出ていると考えられる．

4．今後の課題

今後は，まず対処法等のアシスト情報（ノウハウ）の蓄積を第一に進めることを考えている．

他に必要な機能としては，アシスト情報をプログラマが簡単に変更，追加できるようにする．各プログラマがノウハウを入力することによって，自然に情報が蓄積される．それによって多くの情報を他の人と共有できるようになり，大きな効果が生まれると期待できる．



（a）一つの対処法で解決する場合

（b）複数の対処法から一つの対処法を選択する場合

（c）複数のエラーから対処法が決まる場合

図2．エラーとその対処法の関係

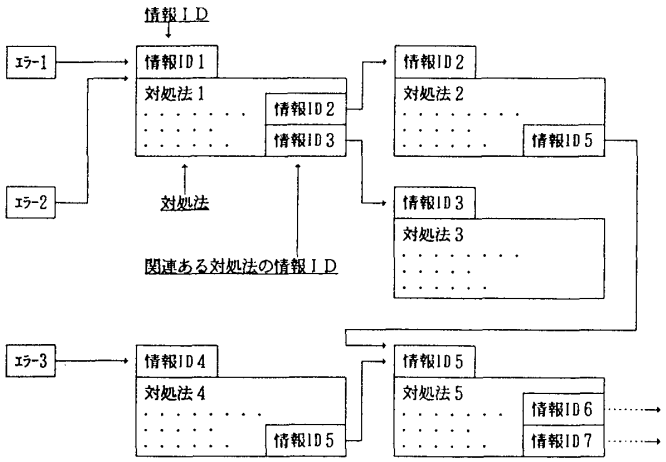


図3．アシスト情報の構造

参考文献

[1]上田・笠原・渡邊:「ﾌﾟﾛｸﾞﾗﾑ構成図をﾍﾞｰｽとしたﾌﾟﾛｸﾞﾗﾑ開発環境」，情報処理学会37全大(1988)

[2]橘・上田・渡邊:「ﾀｰｹﾞｯﾄに依存しない操作性を持つﾃﾞﾊﾞｯｸﾞ環境の構築」，情報処理学会39全大(1989)