

大規模問題に対応した視覚プログラミング方式*

1 R-1

小池雄一 前田康行 古関義幸

NEC C&C 研究所

E-mail: koike@mmp.cl.nec.co.jp

1 はじめに

アイコンックプログラミングは、図形で表現されたプログラム部品同士を線で結んで関係を定義し、プログラムを作成する技術である。アイコンックプログラムには、可読性が高い、プログラムのカスタマイズや保守が容易である、という特長がある。しかし、これまで画像処理、数値計算[1]といった特定の分野を除いては、実用的なアプリケーション(以下 AP)の開発に用いられることは少なかった。その最大の原因としてスケーラビリティ問題が挙げられる。これは、対象 AP が大規模・複雑になると、プログラム中の図形の数、線の交差・ループの数が増加し、可読性が極端に低下する問題である[2]。アイコンックプログラムの最大の特長である可読性が失われるため、保守性、カスタマイズ性等が低下し、大規模・複雑な AP の開発が非常に困難となる。

本研究の目的は、大規模・複雑な AP の可読性の向上である。そして、大規模・複雑問題を、規模と複雑さの二つの観点から捉え、それぞれ、モジュール化技術、部品マルチビュー技術を用いて解決可能であることを示す。また、これらの技術をビジネス AP 向け汎用アイコンックプログラミングツールである HOLON/VP 上に実現し、手法の有効性を確認する。

2 アイコンックプログラムの可読性

2.1 可読性の低下要因

AP が大規模・複雑になるとアイコンックプログラムの可読性は低下する。可読性を低下させる要因としては以下のものが挙げられる。

A) 図形の数

プログラム中に含まれるアイコン、線の数が多くなると、プログラムの内容を理解することが困難になり、可読性が低下する。

B) 線の交差・ループ

線が複雑に交差していたり、ループになっていると、データや制御の流れの理解が困難になり、可読性が低下する。

C) プログラムの意味と幾何的な配置の不一致

プログラム部品の幾何的な配置場所がプログラムの意味的な関係を反映していない場合、可読性が低下する。例えば、一連の処理を結ぶ線がプログラム画面の端から端に飛んでいるような場合、その処理を理解することが困難になる。

これらの要因のうち、A)は規模に関する要因であり、B), C)は複雑さに関する要因である。大規模プログラムの可読性を向上させる手法としてはモジュール化が存在する。しかし、プログラムが規模だけでなく、複雑さの要因を多く含んでいると、従来の手法では可読性を向上させることは出来ない。例えば、図1のアイコンックプログラムは、線の交差やループを多く含んでいるため、モジュール化で可読性を向上させることは出来ない。従って、大規模・複雑問題に対応するには、複雑さの要因を解決する技術が必要である。

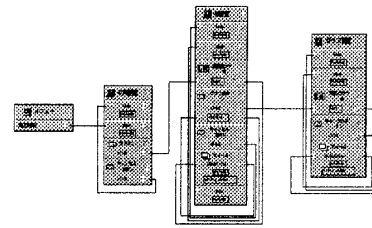


図1 複雑さの要因を含んだプログラム

2.2 本研究のアプローチ

2.1で述べた可読性低下要因のうち A)はモジュール化の手法で対応可能である。しかし、B), C)には対応できない。そこで、本研究では、部品マルチビュー技術(以下 MVO: Multiple View Object representation)を提案する。通常、アイコンックプログラミングシステムでは、プログラム部品の一実体が一アイコン図形に対応する。これに対し MVO では、一実体に対して、複数アイコンを対応させられる。これにより、アイコンックプログラムのレイアウトの自由度が増す。この自由度をユーザが適切に利用することで、可読性低下の要因である、線の交差やループの数を減らすことが可能となる(図2)。

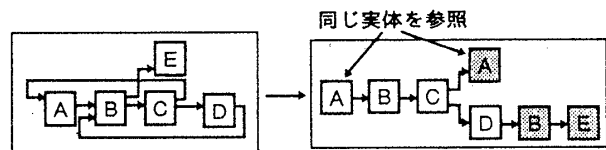


図2 MVOによるループの削減

つまり、MVO 利用により、複雑かつ大規模なプログラムを、大規模ではあるが単純なプログラムにすることが出来る。これに、モジュール化の技術を組み合わせることで、複雑かつ大規模なプログラムの可読性を向上させることが可能となる。

* A Visual Programming Technique for developing large scale applications; Yuichi Koike, Yasuyuki Maeda, and Yoshiyuki Koseki; C&C Research Laboratories, NEC Corporation.

	小規模	大規模
単純	-	モジュール化により可読性向上
複雑	MVOの利用で可読性向上	MVOにより大規模・単純なプログラムに変換

3 可読性向上のためのプログラム記述方法論

2.2節で述べた技術は、レイアウトの自由度を提供するものであり、ユーザがそれを利用して可読性を向上させる必要がある。従って、可読性を向上させるためのアイコンックプログラム記述方法論を合わせて示すことが重要と考えられる。本稿では、筆者等が開発したビジネス AP 構築システム HOLON/VP を例にとり、アイコンックプログラムの可読性を向上させるための方法論を示す。

a) 処理種別による分割

通常、アプリケーションはいくつかの処理から構成されている。このような場合、プログラムを処理に対応した固まりに分割して記述すると、可読性を向上させることが可能となる。例えば、図3-a)は GUI ベースの DB アクセス AP であるが、画面遷移と DB アクセスの二種類の処理によって構成されている。この場合、MVO を利用してプログラムを図3-b)のように処理種別毎のクラスタに分割することで可読性が向上する。

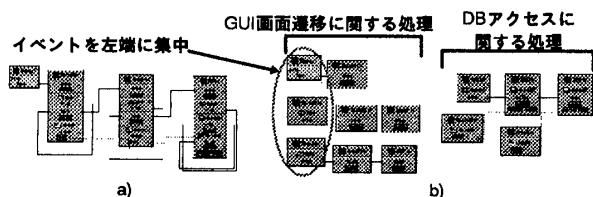


図3 処理種別による分割とイベントの集中

b) イベントの集中

多くのビジネス AP は GUI 画面中心のイベント駆動型 AP であり、GUI 部品イベントから処理が起動される。このため、各 GUI 画面で定義されたイベントを左端に集中させると、処理の始まりを認識し易くなる。図3-b)では各クラスタの左のアイコンから全処理が開始するため、処理の始まりと処理の流れの理解が容易である。

c) 線の交差とループの削減

以上の2方法論は、プログラムの意味を幾何的な配置に反映して、可読性を向上する手法である。これらの手法で可読性を向上させた後、MVO を利用して残った線の交差やループを削減することで、更に可読性を向上させることが出来る。

d) モジュール化

以上の3方法論の適用により、大規模・複雑なプログラムは、大規模だが単純なプログラムに変換される。これにモジュール化を適用して、一プログラム画面に含まれるアイコンや線の数を減らし、可読性を向上することが可能である。モジュール化を適用する際、通常のテキスト言語と同様、一画面中に含まれるプログラム部品の粒度を揃える等の方法論を用いると一層の可読性の向上が可能である。

4 方法論の適用例

図4に、本研究の手法を適用せずに記述したプログラム a)と、適用したプログラム b)を示す。b)は、

1. モジュール化されていて構造が理解し易い
2. 各モジュールが処理種別に対応している
3. プログラムの意味が配置に反映されている
4. 線の交差やループが少ない

といった性質を持ち、a)に比べて可読性が高い。これにより、本研究の手法の有効性が確認されたといえる。

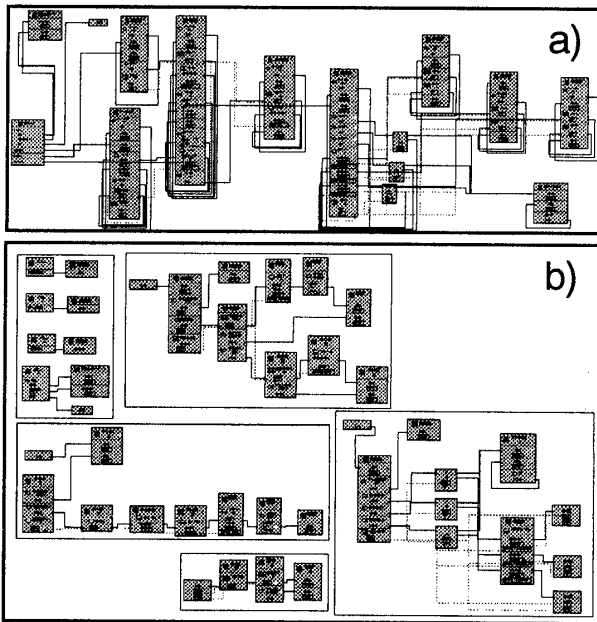


図4 本研究の手法の適用による可読性向上

5 まとめ

本研究では、大規模・複雑な AP をアイコンックプログラミングで扱うには、可読性の向上が最重要と位置づけた。そして、プログラムの可読性低下要因である複雑さと規模を、それぞれ MVO、モジュール化技術を用いて解決可能であることを示した。また、これら技術を適切に適用するためのプログラム記述方法論を合わせて示した。

また、今後の課題としては、一部の記述方法論を自動化し、ユーザによるプログラムの可読性向上を支援するツールの構築、比較実験の実施による可読性の向上効果の定量的な検証等が挙げられる。

参考文献

- [1] Kimura T. D., "Hyperflow: A Visual Programming Language for Pen Computers", *Proceedings of the 1992 IEEE symposium on Visual Languages*, pp. 209-214, 1992.
- [2] Poswig J., Vrankar G., and Moraga C., "Interactive Animation of Visual Program Execution", *Proceedings of the 1993 IEEE symposium on Visual Languages*, pp. 180-187, 1993.