

プログラム可視化システムにおける 動作表示手法

神子 真弓, 相川 聖一, 木村 美奈子, 小野 越夫, 毛利 友治
富士通株式会社

1. はじめに

本システムでは、プログラムの動作を設計図上でアニメーションすることにより、プログラムの理解を助けることを目的としている〔1〕。一般にソフトウェア設計は階層的に行われ、各階層では様々な視点から対象ソフトウェアを眺めることが行われる。そこで、プログラム動作を把握しようとする場合、一つのプログラムに対して、種々の側面から見た複数の設計図上で同時に可視化できる機能、即ち、階層的・多重的表示機能が必要となる。

2. プログラム可視化の基本方式

本システムでは、各設計図ごとにウィンドウを作成し、プログラムの構造・動作をウィンドウ上の図形に写像し可視化を行う。しかし、プログラムからウィンドウへの直接的な写像では、階層的・多重的表示及びウィンドウの表示/非表示等の表示制御を実現することが難しい。そこで、プログラムとウィンドウとの間に設計図の構造を表現する図形構造を設けた。プログラムの構造及びその動作と設計図との対応を図形構造によってとり、その図形構造をウィンドウを通して図形上に写像することでプログラム可視化を行う。

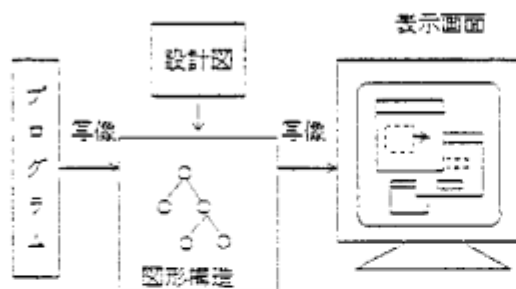


図2-1 プログラム可視化のための写像

プログラムと設計図との対応は、プログラム動作解析部によって行われる〔2〕。本論文では、この図形構造を用いてどのように動作表示を行うかにつ

いて述べる。この際、以下の3点が重要である。

- ① 設計図の形式化
- ② 階層的・多重的表示を可能とする図形構造
- ③ 図形構造からウィンドウへの写像

3. 設計図の形式化

まず、設計図の構成を形式化する必要がある。本システムでは、設計図を木構造と捉える。その各ノードを絵葉と呼び、設計図としての機能を持つ絵葉を特に部分設計図と呼ぶ。絵葉は設計図中の1つの図形に対応し、その図形形状及び形状変化を可能にする操作を持つ。

どのような設計図でも共通に使われるリスト・キュー・アレイ等の基本的な絵葉を、使うたびに定義することは無駄である。そこで、それらの絵葉をあらかじめ定義し絵葉辞書に格納することとした。特に基本的な絵葉についてはシステム定義の絵葉として提供される。各設計図は、絵葉辞書内の絵葉を葉ノードとして階層的に構成され、設計図ライブラリに格納される。

4. 階層的・多重的表示を可能とする図形構造

図形構造は、プログラム可視化開始時に設計図ライブラリを参照して生成される。前述のように、プログラム動作は図形構造の構造変化に写像され、さらに表示画面に写像されるが、この写像方式には次の2つがある。

- ① 構造変化を根から葉へ伝播する方式
- ② 構造変化を葉から根へ伝播する方式

本システムでは、このうち②の方式を採用した。表示画面への写像が設計図のレベルで一元的に管理できる点と伝播すべき情報が少なくて交わる点とその理由として上げられる。動作解析によって得られたプログラム動作を、図形構造内の葉となる絵葉が受け付け、図形構造の構造変化及び画面操作命令の生成を行う。この命令は座標変換等の処理を施されながら親へと伝播される。そして、設計図のレベルに到達した時点で表示画面へ送られる。

部分設計図は、子供の絵画から構築される図形とそれ自身の図形の2つを持っている。そこで、子供から画面操作命令が伝えられた場合、それを交換して親に送るモードと、子からの情報を無視して自身の図形情報を親に送るモードとを持つ。このモード切り換えにより親に送る情報を変化させ、詳細表示と省略表示を行うことにより、階層的表示を可能としている。

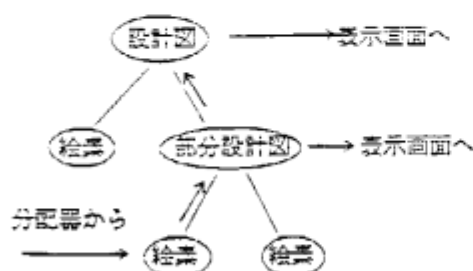


図4-1 部分設計図に関する情報の流れ

また、多重的表现とは1つのプログラムを複数の視点から見た設計図を用いて表現することであり、図形構造上は1つの絵画が複数の親を持つことに相当する。しかし、このような図形構造を作成することは難しいため、現在は設計図ごとに絵画を複写する方式を採用している。

5. 図形構造からウィンドウへの写像

各設計図・部分設計図から表示画面へ送られる画面操作命令は、それぞれ対応するウィンドウへ送られなければならない。また、ウィンドウと対応付いていない設計図についても、対応付けられた際に表示が可能のように内部状態を保持しておく必要がある。図形構造から直接ウィンドウに操作命令を送る方式では、対応するウィンドウの有無を図形構造内で管理しなければならず、図形構造が複雑になって

しまう。そこで、図形構造とウィンドウの間に集配器を設け、そのような管理機構を図形構造から分離した。

また、設計図の表示/非表示、スクロール等のユーザ・インタラクション機能[3]はウィンドウごとに実現する必要がある。そこで、各ウィンドウにそれを管理するアニメータを設けることにした。図形構造から送られる画面操作命令は一旦集配器に集められ、対応するアニメータに分配される。アニメータは送られてきた画面操作命令に基づくグラフィックス命令を生成し、実際にウィンドウへの描画を行う。

6. おわりに

現在、図6-1に示す構成のシステムを試作している。これにより、階層的表示・多重的表现が可能になり、またユーザ・インタラクションの実現が容易になった。ユーザ・インタラクション機能の一つとして、画面の録画・再生機能を考えているが、このための表示履歴管理方式の実現が今後の課題として上げられる。

【謝辞】

本研究は第5世代コンピュータプロジェクトの一貫として行われた。御支援頂きましたICOT第1研究室の方々に感謝致します。

【参考文献】

- [1] 市川他：「プログラム可視化システムの概要」，本論文集
- [2] 石崎他：「プログラム可視化システムにおける動作解析手法」，本論文集
- [3] 相川他：「プログラム可視化システムにおけるユーザ・インタラクション」，本論文集

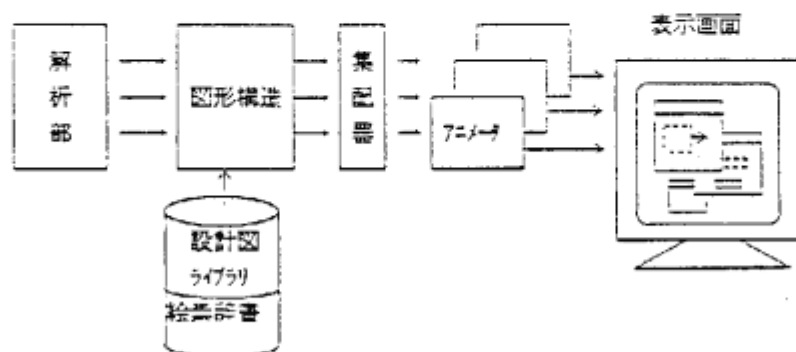


図6-1 システム構成