ソフトウェア設計の視覚多次元化

5F-9

小林 潔

日本アイ・ビー・エム株式会社 サイエンス・インスティチュート

1. はじめに

近年，ソフトウェア開発の生産性・品質を向上させる方法論・ツールが盛んに議論されている。その中で，ソフトウェア・ライフサイクルの各過程を支援し統合しようという試み[1]，費用対効果の観点から，上位レベルを重視しようというアプローチ[2]，グラフィックスを利用しようという試み[3]等々が提起されている。一方，下位過程では機械化しやすかった線形言語パラダイム(Linear language Paradigm)も，主要タスクを階層的に分解し細分化したり，他人に設計を伝える時などに，限界を持っている[4]。それ故に，上位過程からの階層構造的情報を，理解しやすい形で下位過程へ伝達でき，利用できるパラダイムが求められている。そこで，我々は，CAD/CAM の経験を踏まえて[5]，ソフトウェアの開発過程においても，図形情報が重要な役割を果たすと仮定する。この仮定を基に，要求分析・概要設計レベルにおける視覚多次元化の枠組およびそれを支援するツールを提案する。

2. 視覚多次元化の枠組

出発点として，Gane流データ・フロー[6]を採用する。その構成要素は，データ・ソース／デスティネーション，データ・フロー(流れ)，データ・ストア，プロセスである。これらを図式化するために，データ・フロー・ダイアグラム(DFD)を用いる。その図形表現としては，図1.に示されたものを用いる。データ・フローの考え方は，様々な利点を持っており，ハードウェアを含めたシステム全体を記述するためにも利用できるが，ここでは，ソフトウェアの記述に主眼点を置く。

ソフトウェアを視覚的に捉えるために，深さ・時間次元を導入し画面上で直接操作できるようにする。深さ次元とは，DFD要素の論理的かつ図形的階層を測る尺度である。これは，全体の複雑さを減らし，矛盾のない自然なソフトウェアの見方ができるようにするために使われる。時間次元とは，データ・フロー，プロセスがどのような順序で制御されるかを示すための尺度である。ここでは，DFDの制御フローを，DFD図の時間的変化と捉えて，アニメーションという形で示す。この2つの次元からの見方を加えたツールによって，要求分析・概要設計は次のように行なわれる。

- DFDのアイコンをピックすることによって，まずトップレベルのDFD図を作図する。既存のデータ・フロー，データ・ストア，プロセスが利用できるときは，システムの図形検索によって該当のものを指定する。
- データ・フローの詳細化を行なう。これは，データの抽象度を落し図形的に構造化することである(図1.参照)。
- プロセスの詳細化を行なう。例えば，図2.のように，プロセスPを詳細化するとしよう。Pには，データ・フローDIとDOが接続されている。この時，ツールは，Pの子の構造PCのビューポート・コンストレイントとして，d1，d2，d3のビューポートicとo1，o2のビューポートocを与える。このコンストレイントをピックし，移動，スケーリングすることによって，PCレベルでのデータ・フローで結ばれたプロセスの組合せを定義する。

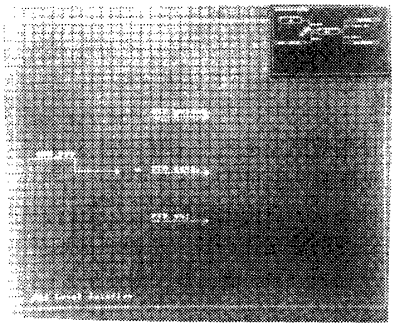

図1. データの詳細化

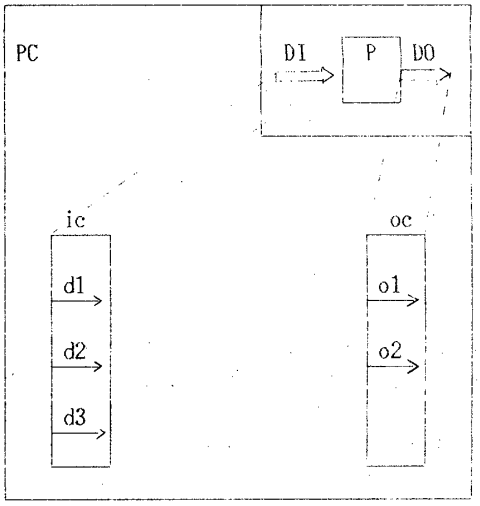

図2. プロセスの詳細化

An Approach to Visual Software Design

Kiyoshi KOBAYASHI
Science Institute, IBM Japan Ltd.

- 各階層ごとに，あるいは，定義されている全ての階層について，DFD図によるシーケンス・アニメーションを行ない，期待される制御および出力データを確認する。例えば，トップレベルで最初のデータをピックしてアニメーションを開始したとすると，それまでに定義された各プロセスの階層に対応した多重ビューポート(depth first)で，シーケンスを色の変化で見ることになる。

プロセスに，階層構造を持たせようというアイデアは，SADT，DeMarcoにも示されている。しかしながら，我々の枠組は，次の点において拡張されている。

1) データ・フロー，データ・ストア，文字もプロセスと同様に階層構造を持て,図形と１対１に対応している。
2) その階層構造を同時に多重ビュー・ポートにマップし，各レベルを，自由に可視化し，直接操作できる。
3) プロセスの詳細化の時，そのプロセスに結ばれたデータ・フローの子の構造を，拘束条件(コンストレイント)として図で与える。
4) 階層構造要素の時間的流れを，多重ビュー・ポートにマップできる。

3. プロトタイプ・ツール

DFDの視覚多次元化を実現するためには，大画面のカラー・ディスプレイが必要となる。今回のプロトタイピングでは，19インチ・カラーディスプレイを持つIBM5080を使用した。

3.1 ユーザー・ビューと直接操作

大部分の操作は，ポインティング・デバイスによってメニュー，アイコン，表示された図形を，ピックすることにより行なわれる。ツールの主な機能は，DFD要素の作図,修正(移動，削除)，属性変更，階層間トラバースと多重ビューポート表示，外部記憶への保存・外部記憶からの読み込みである。例として，プロセスを詳細化する場合を考える。詳細化したいプロセスをピックする。すると，そのプロセスは別のビューポートに別色で表示される。そして，そのプロセスと結ばれたデータ・フローの子構造が作図ビューポートに拘束条件として表示される。そこで再びこのレベルでDFDを書く。次節で述べるような内部構造が作られるので，ユーザーは，作図と同時にデータ構造(設計図構造)全体を画面上で,直接操作しその結果を確認できる。

3.2 内部構造とその表示

すべてのDFD要素とそれに付随した文字の内部構造は，次のようになっている。即ち，任意のDFD要素および文字は，木構造のノードになるとともに(階層構造)，同じ階層レベルでは，どのデータ・フローとプロセスが結ばれているかといった接続関係を持つ。この接続関係には，図形表示のための座標あるいは座標変換行列，および線・色属性を含んでいる。具体的には，DFD要素を示す図形タイプと作図された図形を区別するためのピック識別子によって，親子関係，接続関係を表現する。図形表示は，階層構造を可視化できるように，１レベルを１つのビューポートにマッピングする。現在は，階層表示用に同時に６ビューポートをサポートしている。そのうちの１つは，作図用になるべく広い固定空間を使えるようにしている。その他は，可変サイズで，美学的観点から，縦横比が黄金律比になるようにしている。

4. まとめと議論

ソフトウェアの開発方法を，パラダイム，ヒューマン・インターフェースの観点から捉えて，DFDの視覚多次元化を行なった。この方法は，SADT,DeMarcoのアイデアの拡張になっており，次の利点を持っている。

1) 自然で容易な階層的詳細化が行なえる。
2) データをコール・パラメータとして渡すとすれば，HIPOを使用する時のモジュール構造作成のガイドとなる。
3) Ratenstreich ら[7]が提案している２D-設計法は，自然に吸収でき，垂直水平方向のデータの冗長性を気にしなくてよい。
4) 制御フローの可視化は，システム要求者と設計者のための，システム検証に役立つ。

このような利点を使うことにより，ソフトウェア開発全体の時間・コストの削減ができると期待される。さらに，DFDに固執しなければ下位過程までの拡張も可能である。しかしながら，首尾一貫した拡張にはツールの機能・環境を含めて，なお議論する必要がある。

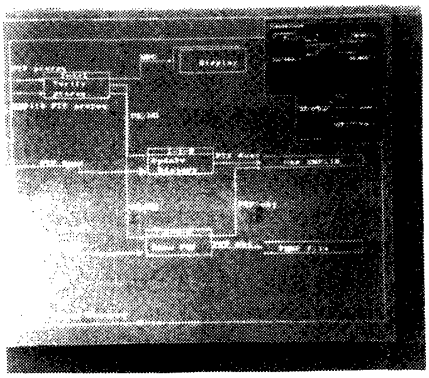

図3. 階層構造のトラバース

参考文献

[1] W.Harrison,IBM Research Report,RC11352(1985).
[2] D.T.Ross,IEEE Trans.on SE,vol.SE-3(1977)16.
[3] For example, see IEEE Computer,vol.18(1985).
[4] W.P.Stevens,IBM Syst.J.,vol.21(1982)162.
[5] S.Uno,K.Shimizu,K.Kajioka and K.Kobayashi, Proc. of IPSJ, (1984)1535,1753.,(1985)2003.
[6] C.Gane and T.Sarson, Structured Systems Analysis, Prentice-Hall, 1979.
[7] S.Ratenstreich et al., IEEE Trans.on SE, vol.SE-12, no.3(1986)377.