

要素モジュールを用いた プログラム構造の設計

5 J-1

齋藤鐵男 上智大学

1. はじめに

システムのモジュール構造設計に欠陥や不確定要因があれば、不確実なモジュール境界がコード欠陥の原因となってシステムの質を低下させる。大量のコードを新たに設計、製作する場合も、モジュール境界の信頼性が保証されれば、モジュールの内部設計以後の工程に、安定した進行が期待できる。

2. DND

CASEには、DFD (Data Flow Diagram) をシステムのモデルとしているものが多く、トップダウンでプロセスを細分化する。トップダウンのモジュール構造設計法には、システムの細部が曖昧な段階でも、上位層から機能の設定と分割を開始できる利点がある。逆にこの上位層で決めた構造が、分割を進めるにつれて矛盾を生じる傾向がある。

ここでは、データをノードとする有向グラフ (DND : Data Node Diagram) をモデルとする方法を提案する。DND法では全主要データに対して、各々を単一の出力とする要素のモジュールを設定する。次に全要素モジュールを設計プログラムに入力し、データをノードとするグラフでシステムをモデル化する。ここではその節点隣接行列を「DND行列」という。以後設計プログラムによるモジュール設計の全過程を通じて、不変のノードにもとづいてグラフの統合処理を行い、機械的に階層化モジュール構造を得る。設計者の介入は、要素モジュールの設定における要素入力を選択に限定される。設計プログラムはモジ

ュール構造設計終了後、実装モジュールのインタフェース部分をプログラム言語で出力し、モジュール機能の実装作業に対して安定した環境を提供する。

3. DND行列の初期化

設計プログラムへの主たる入力要素グラフの出力と入力対で表現されればよい。

$$\text{給与} = \text{支給額} + \text{公課}$$

$$\text{支給額} = \text{基本給} + \text{諸手当} + \text{歩合給}$$

$$\text{公課} = \text{社会保険} + \text{所得税}$$

$$\text{諸手当} = \text{家族手当} + \text{住宅手当}$$

$$\text{歩合給} = \text{受注額} + \text{返品額}$$

$$\text{社会保険} = \text{標準報酬}$$

$$\text{所得税} = \text{支給額} + \text{扶養家族数}$$

などである。これらの要素グラフを共通のノードで結合すると、データをノードとするリニヤグラフが得られる。実際はDND行列の行と列へのデータの割り当てと、行列要素への0/1の格納によりグラフをあらわす。グラフの合成はすべての出力行で、入力列の要素に1、他の要素に0を格納することですむ。

4. 要素記述の欠陥の検出

DND法では、要素記述がシステムのモジュール構造を決定する。要素記述の欠陥はDND行列を調べて検出できる。

①システム入出力の検査 行の和が0のデータ、列の和が0のデータに注目し、システム入出力データの欠落を検査する。

②分離グラフの検出 DNDモデルが分離グラフ構造を示すときは、要素記述の誤りの可能性がある。DNDモデルを無向グラフとみ

なして強連結分解し、これを検出する。

③複数強連結成分の検出 有向グラフにおいて、複数のノードが互いに到達可能な有向経路を持つ場合は有向グラフのままの強連結分解により検出される。正当な強連結成分は、それぞれの強連結成分を1個の代表仮想ノードで置き換える。この操作をここでは強連結縮約と云い、すべての正当な複数強連結なノードを強連結縮約したグラフに対するDND行列を作り、これについて以下の操作を行う。

5. 依存関係による要素の順序づけ

DND行列をデータの依存関係に基づくトポロジカルオーダーに従って並べ替え、上三角ブロック行列が0の行列を得る。

6. モジュールの階層化

オーダリングを終了した行列は、行に対応する要素モジュールを行の順に実行できる。これは図1の設計に相当する。しかし通常は図2の様に多重階層のモジュール構成が多い。

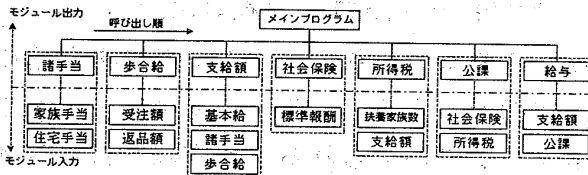


図1 システム構造図(単一階層)

この修正は、次の様に求める。

- (1) DND行列を初期状態とする。
- (2) 行 i 順に上から下へ (3)、(4)、(5)を行う。
- (3) 列 i で要素が 1 の行 j を見つける。
- (4) 行 i の非 0 に対する行 j の要素を 2 とする。
- (5) 行 j の列 i を 3 に変える。

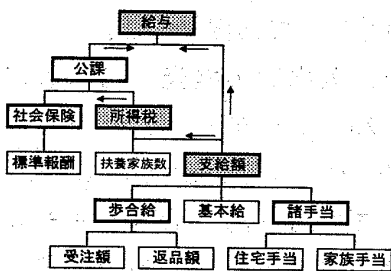


図2 システム構造図(多重階層1)

この行列を参照して、モジュールの行で値3のデータは、自身の局所変数として定義し、これについては該当するモジュールを依存階層順に呼び出す。値1と2のデータは引数で受け取る。

7. 複数呼び出しの問題

図2の矢印で示す様に支給額は所得税と給与に共通の入力である。この種の構造に対するプログラムの実装には2通りの方式がある。

- (1) 呼び出される都度実行する
 - (2) 予め実行し、その出力を保存して使う
- 方式(1)に対しては、6で述べた方法でよい。

方式(2)では、複数の呼び出しを受けるモジュールを、子孫のモジュールと共に、システムの適切な点で実行して結果を保存しなければならない。そのため直接支配ノードに呼び出し点を定める。

図3は図2に対してこの処理を行い、呼出し点を変更した結果の構造図である。矢印は支給額の受け渡し経路を示す。

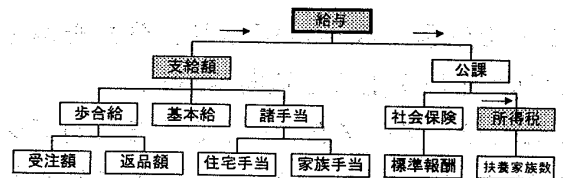


図3 システム構造図(多重階層 2)

8. おわりに

トップダウンのモジュール分解によるモジュール構造設計は、設計者の考察による分解が行われる可能性があり、また細分化が進むにつれて、初期のデータ設計を修正する必要性を来す可能性もある。逆にDNDによる方法は、ボトムアップ法で、殆ど客観的にモジュール構造を設計することが出来る。要素のモジュールは、単一のデータを得るための局所的なデータ依存関係を論理的に定める事になり、設計者の恣意に依存しない。