

モデルのグラフ表現と意味

玉井哲雄

(東京大学 大学院情報学環)

1. グラフによるモデル化

多くのモデル化技法では、なんらかの図式表現を用いることが多い。UML はいつてみれば、これまでソフトウェア開発で使われてきた数多くの図式の集大成というおもむきがある。

図式にはたとえば次のようなものがある。ここには必ずしも分析時のモデル化フェーズだけでなく、概要レベルやプログラムレベルの設計フェーズに使われる図式も含めている。

- a. 制御フローを表すもの(フローチャート, PAD, HCP, SPD, など)
- b. データフローを表すもの(データフロー図, SADT, HIPO, など)
- c. データ構造を表すもの(Jackson 構造図, など)
- d. データ関連構造を表すもの(ER 図, 意味ネットワーク, オブジェクトモデル図, など)
- e. 状態遷移を表すもの(状態遷移図, Statechart, ペトリネット, など)

これらは頂点集合と辺集合からなるグラフ構造をとる点が、共通している。グラフ構造がこれだけよく用いられる理由として、次のような要因が考えられる。

- a. 対象となる世界の「もの」を頂点で表し、「もの」と「もの」との関係を辺で表すことにより、世界がグラフによって自然に表現されること
- b. 「もの」と「もの」との関係には推移律や何らかの推移的な関係が成り立つことが多いが、それはグラフ上の経路の追跡することに対応すること
- c. 描きやすく直観的に分かりやすいこと

グラフの種類に、いくつかのものが考えられる。

- ・ 有向グラフ/無向グラフ
- ・ 木, (有向)無閉路(DAG), 一般グラフ
木構造は階層構造と対応し, 無閉路構造は共通的な下部構造を持つ階層構造に対応する。一般グラフは階層構造を持たない, いわゆるネットワーク構造と対応する。
- ・ 頂点集合の分割(e.g. ペトリネット), 辺集合の分割(e.g. PAD)

グラフの一部をなし, そこに含まれる辺の両端点を頂点集合として含むものを部分グラフという。部分グラフを 1 つの頂点に縮約すれば, その部分グラフをカプセル化, 抽象化したグラフとなる。この操作は抽象の階層を取り扱う自然な枠組みを提供する。実際, データフロー図や Statechart では, この方法が用いられている。

問題は、グラフとして共通の形式をしていても、それぞれで意味づけが違ふことで、それらを混同することがよく見られる。制御フローとデータフローを混同することなどが、その典型である。

2. グラフとモデル上の物理量

グラフは頂点と辺の関係というトポロジーを表している。しかし、グラフを用いてモデル化を行う場合は、頂点や辺に何らかの物理量を結びつけて扱うことが通常である。物理量といったが、ここでは「情報」を相手にしているから、多くの場合は物理量とは呼びにくい。しかし、たとえば電気回路のような典型的な物理を対象としたグラフ・モデルは、多くの示唆を与える。

電気回路の場合、頂点に電位、辺に電流が結びつけられる。辺には電位の差としての電圧も対応する。この物理量とトポロジーとを関係づける法則として、キルヒホフの法則 (Kirchhoff's Law)がある。

a. キルヒホフの電流法則

頂点の周りに流入する電流の総量(流入を正、流出を負として計る)は、0 である。あるいは、電流は各辺の上を、停滞したりとぎれたりすることなく連続に流れる。

b. キルヒホフの電圧法則

閉路(loop)に沿っての電圧の総和は 0 である。

これに関連して、辺の接続に関し直列と並列の 2 種類があるが、それぞれの接続における物理的な性質は、次のように記述することができる (これはキルヒホフの法則から導出される)。

a. 直列： 電流は一定。電圧は加算

b. 並列： 電圧は一定。電流は加算

このような「流れるもの」とそれを駆動する圧力がそれぞれ辺と頂点に結びつけられるモデルは多いが、すべてのモデルがそれにあてはまるわけではない。しかし、キルヒホフの法則のように、

a. 各頂点の周りに出入りする辺全体についての性質を考える。

b. 辺が接続する経路に沿って成り立つ性質を考える。

ことは、基本的である。

3. 代表的なモデル化技法

グラフ構造を用いたモデル化・仕様化技術を、次の 4 種類に分けてみた。何に注目して構造化しているか、という視点からの分類である。代表的な方法論とともに示す。

- a. 事象（動作）の構造： ジャクソン法, Statechart
- b. ものとの関係の構造： ER モデル, オブジェクト指向分析／設計
- c. 情報の流れの構造： 構造化分析（デマルコ法）
- d. 機能の構造： 構造化設計（機能分割), HOS(Higher Order Software)

これらはまた、構造を表す静的なモデルと振舞いを表す動的なモデルとに分類することもできる。静的なモデルと動的なモデルとでは、辺の役割が次のように異なる。

a. 静的なモデル

頂点 A と B を結ぶ辺： A と B がある関係にあることを示す (A から B への有向辺の場合は、「A は B とある関係をもつ」ことを示す)。

b. 動的なモデル

頂点 A から B への辺： A から B へ移動する (制御フロー, 状態遷移など), または A から B へものが流れる (データフロー, ワークフローなど)。

4. おわりに

状態遷移図やデータフロー図の概念を学生に説明すると比較的容易に理解するよう見えるが、実際に図を描かせてみると、とても状態遷移やデータフローを表していないしろものができてしまうことがよくある。グラフ構造は直観的に理解しやすいがために、かえってその意味 (semantics) をあいまいに捉えがちなのであろう。したがって、おなじグラフ構造であるという共通性と、その意味の違いを意識的に考えることが重要である。

これらの図は、長い間に蓄積されたモデル化、設計などのノウハウの結晶であるともいえる。だからたとえば UML も、オブジェクト指向モデルの記述言語というより、そのような広い視野で捉えることが有効ではないかと思う。