

CASEツールにおける方法論の並列融合の構想

2M-1

金子 博 松本 憲幸 岡山 敬 森本 順子
株式会社 東芝

1. はじめに

一般に、従来の統合型のCASEではSA/SD/SPは開発の流れの中で直列的に結合される。しかし、各方法論はそれ自体がかなり柔軟な能力を持つため現実の開発では、必ずしもこの直列的な結合で使用されるとは限らない。

本稿では、SA/SD/SPを直列的に結合する従来のCASEツールに対して、SA/SD/SPを並列的に融合するCASEツールの設計思想に基づいた問題点と実現性について述べる。

2. 現実の適用例が示唆するもの

通常、全工程一貫支援型のCASEツールでは「SA/SD→アルゴリズムチャート→SP」の流れで方法論の直列的な組合せが一般的である。

ところが、本来困難である「構造化分析からのプログラムへの変換」の要求が、かなり上級の設計者からも寄せられる。これは（使用者が要求仕様と実現仕様の混同をしている問題はさておき）、具体的な実現仕様設計に構造化分析を使用しているということになり、プログラム設計手法と構造化分析が相入れる可能性を示唆している。

さらに、構造化分析のモデル全体の適用性とはともかく、そこで使用される個々の設計図は、

- ・ DFD：データ駆動型マシンの並列処理プログラムにも使用される
- ・ STD/STM：オートマトン定義に使用される

など、プログラム仕様記述の能力を持つものであり、その他のCFD/CSPECに至っては、より詳細な制御の流れを定義することが可能である。

また、全く逆にフローチャートを要求モデルの定義などに使用する例もある。

すなわち、

"Concepts for coherence of methodologies in CASE tools." Hiroshi Kaneko, Noriyoshi Matsumoto, Takashi Okayama, Junko Morimoto TOSHIBA Corporation

- (1) 構造化分析の一部あるいは全体がプログラム化される実体モデルの定義に使用できる可能性
- (2) 構造化分析とフローチャート等の定義の親和性等の可能性

があるといえる。

3. 現実の要求への適用問題

2節の(1)の場合、以下のような問題がある。

- ・ 実体化されるモデルの定義で使用される構造化チャート、フローチャート、各種仕様書との相対関係はどうなるのか？
- ・ 処理速度 ∞ 、処理能力 ∞ の数学的モデル解釈の不都合は何処に現れるのか？

2節の(2)の場合は、以下のことが言える。

- ・ フローチャート等で構造化分析モデルをより厳密化できるという可能性と同時に、本来これを使えば欠落を避けられる定義情報がある可能性がある。

特に、(2)に関してはゆゆしき問題である。なぜならば、考える手段がその部分で与えられていない、あるいは、考えたことを残せない/伝えられない可能性が潜在するからである。そういう意味で、SA, SD, フローチャート等を直列結合ではなく並列結合したCASEツールの可能性を探る手がかりとして興味深い。

4. 並列結合モデルの適用性の考察

4.1 SAとFLOW/PADの相対関係

DFD/CFD上のプロセスもフローチャート上の処理ボックスも一般的な処理に対応しており、CFDに付随するCSPECはフローチャート等と同様に処理に関する制御の流れを定義する。

ただし、CSPECによる制御の定義は入力に対する状態遷移図/決定表によって記述されるため、モデ

ルが明確である一方でフローチャートほど一般的な制御の流れが定義できない。例えば、前者では入力が発生した時点の測定結果によって決定される制御の流れはモデルの意味論の範囲内では定義できない。

4.2 SAとSDの相対関係

構造化分析モデルを中下流で実体化されるモデルに変換する場合、従来の上流限定利用モデルで行われていたDFD→構造化チャート変換形態がふさわしくない場合が存在する。

- ・CSPECはCFD平面上のプロセス間での制御の流れを記述するが、この平面上のプロセスのいずれかを他のプロセスよりも上位に置くと、下位から上位への制御が出現して実体化された処理の制御が一つの制御平面で記述されない難解な実装を生じる。

コントロール平面を保存するためには、DFD/CFD一平面が構造化チャートの同一階層となるような変換が必要である。この対応関係では、従来のモデルでトランザクションセンターとして選択されたプロセスが行ったデータの分配は、DFD/CFDにより記述される処理平面を内部処理として持つ上位のプロセスが担当することになる。

4.3 数学的モデル解釈の問題

構造化分析手法から離れた部分での使用要求が最も多いSTDは、オートマトンの理論によって意味付けがされているが、これを現実的な仕様記述として利用する場合、次の問題がある。

(1) 無限大の遷移速度(遷移時間0)の困難

遷移時間0の数学的モデルと有限の遷移時間を持つ現実のモデルとの違いは、後者の場合に入力を受け取った後に次の状態に遷移完了する以前に次の入力を受け取る可能性があることである。逆に言えば、この困難は以下の条件下では回避可能である。

- ・STDにより記述される有限状態マシンの遷移時間の最大値を τ とした場合、このマシンに対する入力を待ち行列化し、 τ の時間間隔を置いて処理するようなシステムを構成する。

(2) 決定的/非決定的解釈の困難

有限数の状態を経由した後終了状態に達した遷移のみを有効とする非決定的な解釈は、各遷移に付随して副作用を生じるような現実的な処理の仕様記述には通常(Prologなどの特殊な処理を除

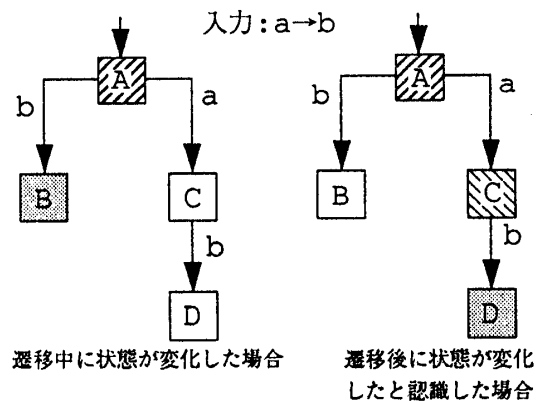


図1 遷移時間を持つモデル

いて)向かない。

決定的な解釈に従ったSTDによって仕様を記述する場合、遷移の定義が決定的であるが故に、ある条件下でのみ起こる処理の記述が困難である場合がある。例えば、マシンが内部的にスタックを消費しながら処理を行う場合、スタックトップに到達すると例外処理を行う必要があるが、これを決定的に記述するには、全てのスタック状態の数だけ状態を用意するという非現実的な記述が要求され、これは数百万回に一回程度の例外処理に備えるために、いたずらに仕様を煩雑化されるに他ならない。

この意味では、STDによる詳細設計には限界があると言える。

5. おわりに

従来の方法論の直列的な配置の問題点を解決するために、方法論の並列的な融合を行うことは、それぞれの方法論の適用範囲が変化すると同時に、ここで説明したような方法論間での意味論の対応が重要な問題となる。方法論の有効な適用方法を探り、より現実的なCASEツールの研究にはこの様な問題への対応が必要である。

参考文献

- (1) Hatley他:リアルタイム・システムの構造化分析,日経BP社,1989
- (2) Aho他:Data Structures and Algorithms,ADDISON-WESLEYPUBLISHING社,1987
- (3) マーチン他:ソフトウェア構造化技法,近代科学社,1986
- (4) 金子他:設計方法論に基づいたCASEツールの落とし穴,情報処理学会第49回全国大会講演論文集,5-91
- (5) 原田他:CASEのすべて,オーム社,1991