

MTAとCASTによるフォーマルアプローチ開発

日大生産工(院) ○竹田 信夫 日大生産工 齋藤 敏雄

1 まえがき

現在の情報システムの開発においては、ウォーターフォールモデルと呼ばれる開発手法が一般的である。この開発手法はシステム開発における作業を段階ごとにまとめ、各段階を逐次的に処理していく手法であり、特に大規模なシステムの開発に向くとされている。

しかしながら、ライフサイクルモデルは各段階の処理の結果が次の段階以降に行かないと実現しないため、しばしば現実の開発の現場においては問題が発生している。

このため、軍事や航空宇宙産業等極めて高度な信頼性と正確性が必要とされる場面においては、特に仕様策定の段階にで、数学的あるいは論理的な記述手法をもちいた正確でかつ厳密な記述を行うことにより、問題が発生する可能性を極小にする開発手法が提案され、実施されてきた。こうした開発手法をフォーマルアプローチという。

フォーマルアプローチは、歴史的には主として軍事用途や組み込みシステム、交通システムなどに用いられてきたが、次第に適用される範囲を広げ、現在では、高度化した経営情報システムの開発においてもその適用が模索されている。

本報告では、フォーマルアプローチ開発の中でも、経営情報システムの開発に有望な開発手法としてMTAとCASTを用いた開発について報告する。

2 フォーマルアプローチ開発

以下ではまず、フォーマルアプローチについて説明したい。

現在使われているフォーマルアプローチの流派は大きくいって、特定目的用のものと、汎用的な開発に利用可能なものに大きく分けることができる。前者の例としては、回路設計仕様の記述に特化したSpecCや通信プロト

コルの仕様記述を主たる適用分野とするLOTOSがある、後者の例としてはVDMやZが有名である。

厳密に言えば、フォーマルアプローチとは形式記述言語を用いて、仕様や機能の記述を行い、その検証や矛盾の排除を図る手法であり、記述に用いる言語はその手法の一部をなすものである。

例えば、VDMはIBMのウィーン研究所で開発された、フォーマルアプローチの代表的手法であり、仕様記述言語はVDM-sl

(Specification Language) , あるいはVDM++を用いる。

VDMは組み込みシステム等を中心として広く利用されており、日本ではフェリカ等の開発でその有用性を示したとされる。

現在、フォーマルアプローチを汎用的なシステム開発で用いる場合にはいくつかの問題が指摘されており、利点と欠点を考慮した上で利用することが望まれる。

例えば、日経コンピュータでは、現時点におけるフォーマルアプローチを採用する際の利点、欠点を、VDMを利用したいくつかの事例をもとにまとめている。本報告ではこのまとめを手掛かりとして、フォーマルアプローチを採用する際の問題点について検討したい。

利点

1) 厳密な記述による仕様の漏れ、矛盾、外部仕様との不一致等の検出が可能

フォーマルアプローチではその記述が複数の解釈が生じようのない厳密な記述ができ、また、複数の仕様記述の比較検証を行うことで、仕様の漏れ、矛盾、外部仕様との不一致等の検出が可能である。

自然言語を用いた記述の場合、その解釈にずれの生じる余地があり、また、形式的な操作によって比較検証することが難しいため、仕様の漏れ、矛盾、あるいは外部仕様との不

一致等の検出は人が逐一確認する必要がある、間違いが混入する余地がある。

2) バグの発見

集合論的モデル記述言語で記述されたモデルと計算機言語で作られるコードの振る舞いの差を形式的操作によって見出すことができるため、テストの実施によって発見できなかったバグも取り除くことができる。

3) テスト項目の洗出し

仕様から必要なテストの項目について任意の時点でその項目のリストを決定することができる。

欠点

1) 記述の困難性

記述が困難といった場合、問題はさらに2つに分けられる。

第一の批判は、通常のシステム開発者にとってそもそもフォーマルアプローチの記述言語の習得が難しいという批判である。

第二の批判は現在一般的に使用されている形式記述言語における記述量が多いという点である。

2) 記述が二度手間になる。

現在一般的に使用されている形式記述言語を用いたフォーマルアプローチでは、確定した仕様書を記述する制限された自然言語、実装のための計算機言語とフォーマルアプローチの記述言語の3つを使うことになり、作業的に純粋にステージが1つ追加されたのと同様のこととなり、特別の目的以外ではコスト面等から使用しにくい

3) 形式的な検証における多大な負荷

現在一般的に使用されている形式記述言語を用いたフォーマルアプローチでは、仕様決定後仕様の検証を行う。このプロセスにより、バグの発見等が行えるのであるが、このプロセスの技術的あるいは作業量的負荷が大きい

このため、各フォーマルアプローチではそれぞれ欠点取れた問題を修正するための対応が行われている。

例えばVDMでは一般に使われている仕様記述言語VDM-slが独特の記述を用いて記述すること、VDM-slの記述が、基本的に仕様の明確化が目的であるため仕様記述の2重化が問題となることなどについて対応するため、VDM++というオブジェクト指向の記述言語を開発し、さらに各種ツールにより実行コードを生成が可能としている。

VDM++はJAVAやC++等の現在よく利用される開発言語と近い構文を持っている。

このため、集合論をベースとしたVDM-slの記述の方法よりもシステム開発者の理解と習得が容易になっている。

また、VDM++は各種ツールにより、仕様上の問題点の確認や実行コードの生成等が行えるため、各種ツールを利用できるリソースが存在することが前提ではあるが、記述の簡素化、検証プロセスの負荷軽減といったことがはかれるようになっている。

3 MTAとCASTを用いたフォーマルアプローチ開発

本報告では経営情報システムの開発に有望な開発手法としてMTAとCASTを用いた開発について報告する。

モデル理論アプローチ(Model Theory Approach, 以下MTA)は、フォーマルアプローチによるシステム開発の方法の1つであり、一般システム論を基礎としたフォーマルアプローチ開発である。

MTAはシステム論を背景としたモデル記述とシステムアプローチによる問題記述を行うことにより、システムの開発を行う。システムの記述にはCAST(Computer Acceptable Set Theory)を用いる。

CASTによって記述されたモデルは、単に開発するシステムを表現しているのではなく、ユーザのシステムの問題をモデル化したものである。

このため、現実の開発されるシステムとモデルは、VDM-slで記述される外部機能と仕様のように1対1の対応を持たない。作成されるシステムは図1に示すように問題解決システム、あるいは目標追求システムを持つ実際の情報システムとなる。

システムの実装は、目標記述によってシステムを開発する、第4世代言語と問題解決のエキスパートシステムによって行われる。

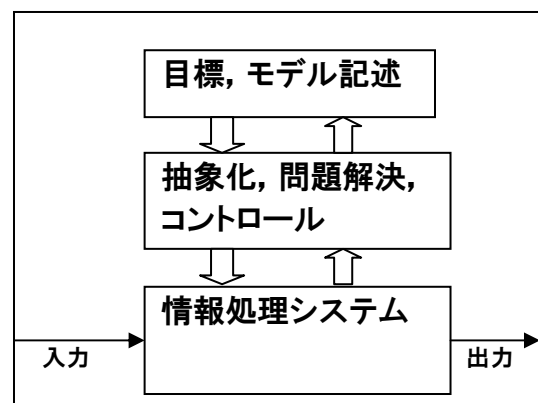


図 1 MTA によるシステム開発の概念図

こうした、システムの原型は1980年代に提案されていたが、当時としては過大なシステムを要求することもあり、普及しなかった。

しかしながら、現在の情報処理の技術的環境においては過大ではなくなっている。

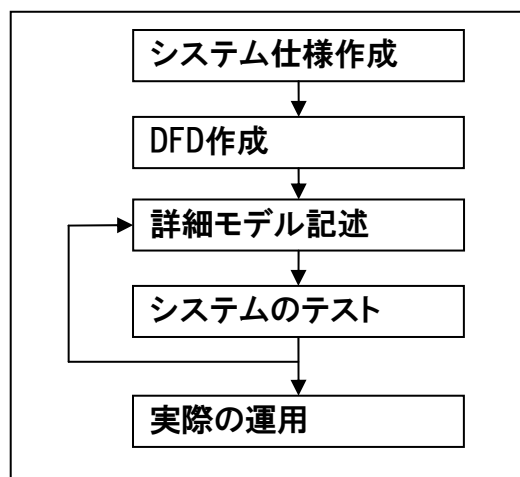


図 2 MTA によるシステム開発の全体的流れ

実際のMTAによるシステム開発は図2のような流れによって行われる。

全体の開発はプロトタイプ型の開発となる。

このため、全体的なシステムの適応対象やパフォーマンス、スケーラビリティといった問題については弱点となりうる。

他方、プロトタイプ型の開発はユーザの満足度を高めることができる。また、MTAは、プロトタイプ型の開発の問題点である開発時間の遅延をシステムの自動生成によって大幅に短縮できる。

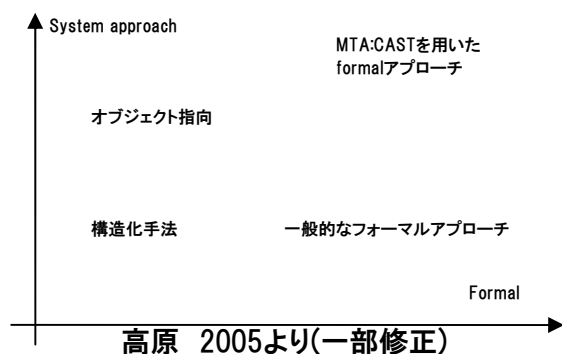


図 3 MTAとCASTによるフォーマルアプローチ開発

MTAとCASTによるフォーマルアプローチ開発を既存の開発方法論と比較すると、図3にあるように、フォーマルアプローチ開発であり、かつシステムアプローチであるという点に大きな特徴がある。

フォーマルアプローチであるとは、数学的に厳密な仕様記述ができることである。

前節に述べたように、フォーマルアプローチはシステム開発における問題の回避がはかれる。

MTAの記述言語であるCASTは、名前の通り

数学の集合論のサブセットであり、

1) 厳密な記述による仕様の漏れ、矛盾、外部仕様との不一致等の検出が可能

2) バグの発見

3) テスト項目の洗出し

という形式手法の利点を全て満たすことができる。

他方、CASTによる記述は、

・フォーマルアプローチの記述言語としては平易であるため、長期間のトレーニングを必要としない。

・4GLや、問題解決システムを利用できるため、複雑なシステム状態の記述や問題解決の記述が容易で、実際に問題解決システムを生成することが可能。

・実システムを生成できるため、GUIの使い勝手といった、感性に関係することについてもチェックできる。

という特徴をもっている。

この特徴により、前述したフォーマルアプローチの欠点とされる2つの「記述の困難性」は大きく軽減されている。

また、システムの自動生成によるプロトタイプ型の開発としたことにより、「記述が二度手間になる。」「形式的な検証における多大な負荷」という問題も避けることができる。

システムアプローチであるとは、システムの記述に際して、システム論に基づくシステム概念とシステムモデルが用いられるということである。

システムアプローチであるかないかは、開発手法自体の優劣を示すものではないが、システム概念やシステムモデルを持たない開発は、情報システムの仕様作成がその情報システムの振る舞いの記述に限定されてしまうので、そのシステムのユーザからみた合目的性等は保証されない。

また、サブシステム間の関係やつながりについて、構造的な関係を明示的に表現するにはシステム概念やシステムモデルによる記述が優れている。

CASTはシステム概念やシステムモデルでしばしば用いられる集合論のサブセットであるため、システム概念やシステムモデルの記述について、実績があり、表現方法も確立している。

このため、システム概念やシステムモデルの記述際には、個別の開発方法論特有の概念や表現方法を理解する必要がなく、この意味で習得が比較的容易であると言える。

また、前述した、4GLや、問題解決システムを利用と、集合論的表現は極めて適合

性が高い。

4 まとめ

本報告では、現在MISでも適用が始まっている形式手法による開発の状況について説明し、特にMTAとCASTを用いた開発について説明した。

MTAとCASTによるフォーマルアプローチ開発は以下の特徴がある。

- ・システムアプローチであり、かつフォーマルアプローチである。

- ・プロトタイプ型の開発を行う。

- ・4GLや、問題解決システムを利用できる。

MTAとCASTによるフォーマルアプローチ開発は多くの点で現状の問題点を改善しているため、更なる適用範囲の拡大が望まれる。

「参考文献」

- 1) 「脚光を浴びる形式手法」『日経コンピュータ』，日経BP社（2006. 7. 24）
- 2) 旭 貴朗ほか著「経営情報システム開発のためのモデル記述言語:CAST」，経営情報学会誌
- 3) 玉井哲雄著『ソフトウェア工学の基礎』，岩波書店（2004）
- 4) 高原康彦 「経営情報システム開発へのモデル理論的アプローチの実用化」
2005
- 5) 高原 康彦 他著 「形式手法モデル理論アプローチ」日科技連 2007