6-22

情報システム開発のための形式的記述言語 CAST について

日大生産工(院)　○前川　徹　日大生産工　柴　直樹

1. はじめに

本研究では、形式的記述言語である CAST について、その成り立ちや特徴を踏まえた上で新たな情報システムの構想・構築を行う。最終的には構想物を本論で主張する手法で構築することが可能か検証する。また、本手法を他の手法とプログラム量や構築期間などで比較することにより本手法の実用性や有用性などを探ることが目的ではあるが、本稿はそのための予備的な研究報告である。

本稿では、2 節で形式手法の説明と形式的アプローチであるモデル理論アプローチについての説明を行う。3 節で形式的記述言語である CAST とそのアプローチについて説明を行う。4 節で今後の課題を述べる。

2. 形式手法 モデル理論アプローチ[1]

2.1 形式手法

形式手法(Formal Method)は、モデルを記述するときに、自然言語や図、グラフなどを使うのではなく、数学の言葉を用いて記述を行う。数学の言葉を用いることにより、表現のあいまいさを排除して複数の解釈が生じないようにすることが可能となる。逆に、モデルを記述するときに、自然言語や図、グラフなどを用いる手法は非形式手法という。

形式手法が提案され始めたのは 1970 年代であり、有名なものには 1996 年に ISO にて標準化された VDM-SL（Vienna Development Method - Specification Language）や 2002 年に ISO にて標準化された Z がある。

VDM の例ではあるが、ウォーターフォールの概念から形式手法を用い、下流工程からの手戻りを回避するために上流工程で形式手法を使い、見積もりに対して 3 倍の生産性を得たという日本フィッツ(現 CSK システムズ)の発表がある。他にも、フェリカネットワークスによる「モバイル FeliCa IC チップ」ファームウェア開発にも形式的アプローチが用いられた。

2.2 システムアプローチ

システムアプローチは、対称に対してシステムモデルをもち、システム概念及び、システムモデルを使ってモデル化を行う。システムモデルの一つとしてはオートマトンシステムがあり、後述する MTA はオートマトンシステムモデルを使用するシステムアプローチである。

2.3 モデル理論アプローチ

モデル理論アプローチ(Model Theory Approach：MTA)は、モデルの記述を行うときに、数学的な言葉、特に集合論を使用する形式的アプローチである。

MTA の対となる非形式的アプローチでは、いくつか指摘されている問題点がある。問題点としては、未熟なコード化を行うことにより、結果として手戻りコストが大きくなることがあげられる。モデル記述で自然言語や模擬コードによる記述を行うと信頼性は低くなり、それを補うための見直し、検閲が必要になる。それに加えて開発者の誤解が生じる可能性などもある。また、エラーの検出や顧客からのフィードバックの遅さなどもある。

CAST：A formal description language for information system development

Toru MAEKAWA and Naoki SHIBA

形式的アプローチでは非形式的アプローチに対して、以下の利点を持つとされている。
①仕様の矛盾・欠陥を仕様作成時に発見しやすくなる。
②形式的表現により記述の信頼性が高まる。
③手戻りの回避が可能となる。
④機能の追加が容易、モデル自体が文書として使える。

よって、形式的アプローチである MTA はこれらの利点を有している。また、MTA が従来の形式的アプローチと異なるところとして、システムアプローチであるということがある。

図 1 は形式的アプローチをシステムアプローチと非システムアプローチの二つで縦に割り、アプローチの種類を分類したものである。

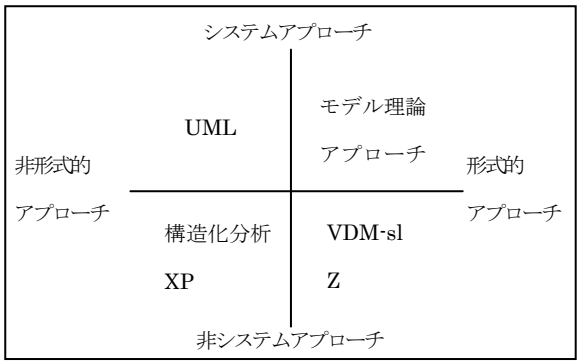

図 1. 他のアプローチと MTA の比較

3. 形式的記述言語 CAST

3.1 CAST とは

コンピュータ可読集合論(Computer Acceptable Set Theory：CAST)は、システムアプローチを持つ形式的アプローチである MTA のシステム記述に用いられる言語である。CAST は形式的アプローチを用いるため、記述には述語、集合、関数などの数学を使い、システムアプローチを用いるため、オートマトンシステム使う。

CAST は形式手法の特徴と、システムアプローチでもある MTA 特徴を持つが、モデルの記述を行うために必要な知識としては、高校レベルから大学初年度の数学能力で記述が可能になることを目的の一つとして開発されている。

3.2 実行環境

CAST 言語の開発・実行環境としては現在 MTA-SDK(Model Theory Approach-System Development Kit)[2]というものが存在する。

MTA-SDK は、CAST 言語で記述されたモデルを extProlog で処理して結果を X-Window システム上で表示するものである。本環境の問題解決システム(Problem Solving System：Solver)では CAST で記述された対象問題モデルの実行を行う。同様に、業務処理システム(Transaction Processing System：TPS)では CAST で記述した TPS の構築と実行を行うことができる。

4. おわりに

形式的記述言語である CAST を用いることにより、形式的アプローチとシステムアプローチの両方の特徴をもった開発が行えることが分かり、その開発・実行環境である MTA-SDK が配布されていることも分かった。

今後の課題としては、参考文献[1]の TPS の例である酒問屋のシステムを参考に、エントリーフォームを TPS に関連させたシステムの構想を考える。そして、構想するシステムが MTA で構築可能か検証したい。また、他の 手法で構築されたシステムとプログラム量や構築期間などの比較を行うことにより、本手法の実用性や有用性を探りたい。比較の例としては、過去にエントリーフォームを導入した LAPP による業務支援システムの構築[3]をしたことがあるため、まずはそれらとの比較などを行いたい。

参考文献

[1] 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

[2] モデル理論アプローチによる 経営情報システム開発 (http://www.geocities.jp/kisoten/) 最終アクセス 2010 年 10 月 28 日

[3] 前川 徹　「LAPP による学園祭運営支援システムの開発」日本大学生産工学部 2009 年度管理工学科卒業研究　2010 年 2 月