第 36 章　フォーマル・メソッド

「フォーマル・メソッド」とは何か

インターネット上の百科事典であるウィキペディアには、「形式手法（フォーマル・メソッド）」について次のように記述されている[1]。

「ソフトウェア工学における数学を基盤としたソフトウェア（中略）の仕様記述、開発、検証の技術。」

今ではコンピュータは、「コンピュータ（計算を行う機械）」というよりは「インフォメーション・プロセッサ（情報処理を行う機械）」と呼ぶ方がふさわしい状態になっている[2]。しかし初期の頃コンピュータは数学と深いつながりを持ち、文字通り計算を行うための機械だった[3]。フォーマル・メソッドには、その頃の名残がある。

フォーマル・メソッドの使用実績

フォーマル・メソッドが寄って立つ学問領域は「数理論理学」であり、数理論理学は独特の記号を使って論理を記述する。記述された論理は全く矛盾や誤解のないものであっても、この記号は特別の教育と訓練を受けた人以外はなじみにくいという欠点がある。ソフトウェア技術者でもこの教育と訓練を受けた人は少なく、一般のユーザのレベルでは多分皆無に近い。そのため記述された要件定義を、ユーザを含めた多くの人でレビューするということがたいへんに困難である。これが、フォーマル・メソッドの使用実績の少なさの 1 つの要因になっている。

フォーマル・メソッドはたいへん品質の高いソフトウェアを生成する方法であり、宇宙工学、原子核工学、航空機工学など、そのソフトウェアに高品質を要求する分野ではいくつかの使用例が報告されている。しかし一般の企業内での情報システムでの使用は、前述の通りたいへんに少ない。

フォーマル・メソッドによる開発の方法

フォーマル・メソッドによる開発には、いくつかの方法がある[Wiki301]。しかしその主な方法は、VDM と Z 言語による開発である[ORegan14]。

VDM（Vienna Development Method）とは、1960 年代から 70 年代にかけて IBM のウィーンの研究所で開発された方法である。その後 VDM はオブジェクト指向化されて、それは VDM++と命名されている[ORegan14]。また VDM は ISO などによって、ISO/IEC 13817-1：1996 として国際標準になっている[4]。

Z 言語は形式仕様記述言語で、次の 2 つのことに焦点を当てている[Wiki301]。

---

[1] いつも引き合いに出している ISO などによる用語集（ISO/IEC/IEEE 24765：2010）には、“Formal Methods”についての記述はない。ただし、“Formal Design”とか“Formal Specification”など、フォーマ・メソッドの個々の要素については多くの記述がある。

[2] インターネット上の情報の検索などを行うコンピュータは、「計算機」というよりは、まさに「情報処理を行う機械」と呼ぶにふさわしい。

[3] 最初のコンピュータは、砲弾の弾道計算を行うための機械だった。

[4] ISO などによる国際規格は原則として 5 年の一度見直しが実施され、改訂／廃止などの対象になる。しかしこれらの規格はこの原稿を書いている時点（2016 年（平成 28 年）8 月）で発行から 20 年近く経過しているにも関わらず、まだ「有効」である。

- コンピュータ・プログラムの簡明な仕様の記述
- 意図するプログラムの振る舞いの証明の形式化

この言語は、1970 年後半にジーン・レイモンド・アブリアル（Jean-Raymond Abrial）等によってイギリスのオクスフォード大学で開発され、後に IBM のパッケージである CICS の開発で使用された。Z 言語もオブジェクト指向化されて、それは Z++と呼ばれている[ORegan14]。さらに Z 言語も ISO/IEC 13568：2002 として、国際標準になっている。

フォーマル・メソッドの限界

前記の要件記述の表現の難しさに加えて、今のフォーマル・メソッドには次の 3 つの限界が報告されている。

その最初のものは、記述された要件に矛盾がないことを証明するために、要件定義が完成した段階でツールによる「検証」を行う。数理論理学ではこの検証は、全てのケースを網羅して行うが、このため検証のケースが多くなってたいへん時間がかかるという問題が出ている[Kisi16]。全数検証ではなく部分検証という方法もあるが、それではフォーマル・メソッドを採用した意味が半減することになる。

限界の 2 つめは、記述されたことの間に矛盾などがないことは明らかにでき、どの部分が適切ではないということも明らかにできる。しかし要件が漏れた場合、つまり記述されていない部分があった場合、「記述漏れがある」との指摘ができないとのことである。

そして 3 つ目は、フォーマル・メソッドによる開発は構造化技法などによる一般の開発と比較して、高くつくとのことである。従って、多額の開発費用を正当化できる領域での使用が適切である[Kisi16]。

フォーマル・メソッドの将来

前述の通り、今のフォーマル・メソッドにはいくつかの限界がある。しかし人工知能（AI）の進歩によりこの限界が取り除かれて、フォーマル・メッソドがソフトウェア工学の本命に変わる可能性がある。

今の限界の最大のものは VDM とか Z 言語とか、ある意味で特殊な言語で仕様を記述しなければならないことにある。しかしこれはそのうち、仕様を自然言語（日本語や英語など）で記述して、必要なら AI がそれを別の必要な言語に変換することが期待できる。

さらに今のフォーマル・メソッドでは、仕様の「漏れ」を指摘できない。しかしこれも、「仕様には本来こういう内容が書かれているべきもの」という要件定義の「常識」をツールが取り込んで、必要と思われる事項の記述がなされていない場合には、「これが書かれていないけれど、大丈夫でしょうか」といった問いかけができるようになるかもしれない。

仕様の検証に時間がかかることとか、ソフトウェア開発が高価につくといった問題は、コンピュータの能力の向上で解決できるだろう。

こういうことで今の限界が解消できれば、超高速開発[5]と並んでフォーマル・メソッドはソフトウェア工学の本命になり、保守を含む広い意味でのソフトウェア開発の生産性を高め、ソフトウェアの信頼性も向上させて、ソフトウェアの世界が様変わりする可能性がある[6]。

私はいま、それを期待している。

---

[5] 超高速開発については、第 28 章で述べた。

[6] ソフトウェア工学の将来については、第 57 章で述べる。

キィワード

フォーマル・メソッド、形式手法、数理論理学、VDM、Z 言語、形式仕様記述言語

略語

VDM：Vienna Development Method

規格

ISO/IEC 13817-1：1996、ISO/IEC 13568：2002

人名

ジーン・レイモンド・アブリアル（Jean-Raymond Abrial）

参考文献とリンク先

[Kisi16] 岸知二、野田夏子著、「ソフトウェア工学」、近代科学社、2016 年.

[ORegan14] Gerard O'Regan, "Introduction to Software Quality," Springer International Publishing, 2014.

[Wiki301] 「形式手法」、https://ja.wikipedia.org/wiki/形式手法
（確認日：2016 年（平成 28 年）8 月 8 日）.

（2016 年（平成 28 年）8 月 8 日　新規作成）
（2017 年（平成 29 年）2 月 13 日　一部追加）