新・組み込みソフトへの数理的アプローチ
～形式仕様記述をどのように使うか～

第10回

要求のトレーサビリティとモデリング
――仕様書の書き方とCBMC

藤倉 俊幸

はじめに

一般にソフトウェアの開発は，要求，設計，実装，テストの順で行程が進む．要求行程では要求獲得，要求分析，要求検証，要求の記述が行われる．

しかし，要求行程における基本的な作業である獲得・分析・検証・記述は，「要求」を「やりたいこと・やらねばならないこと」と解釈すれば設計工程や実装工程でも必要な作業である．たとえば，設計に対する要求とか実装に対する要求という言い方も可能である．JSD法（Jackson System Development）のM.Jackson氏は，「要求はシステムの外にある現象について述べたもの，プログラムはシステム内の現象について述べたもの」と定義している．そして境界に存在するものを仕様と呼んでいる(1)．場所が異なるだけで，現象について述べているという点では要求もプログラムも同一であるということだ．

さらに，プログラムの中を詳細化すれば設計と実装になる．設計はシステム内の現象のうちモジュールとかクラスとか関数によってシステム内の現象を述べたものであり，実装はソース・コードによってシステム内の現象を述べたものと考えられる．そうすると，場所と何を使って述べるかの違いはあるが，**図1**に示したように獲得，分析，検証，記述を通してずっとつながっているものが見えてくる．これはソフトウェア開発の工程におけるトレーサビリティの重要な側面だろう．

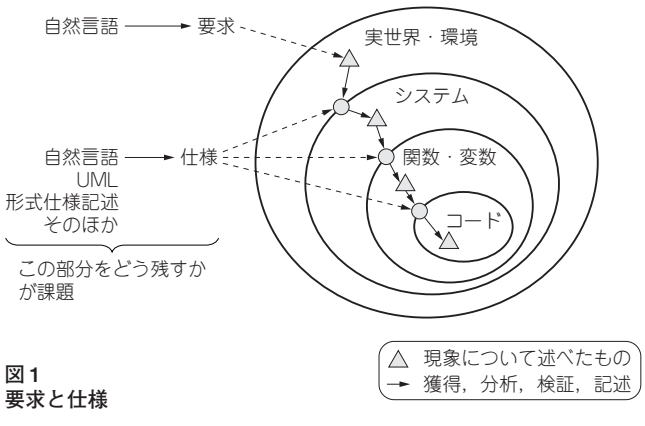

図1
要求と仕様

1 自然言語のデメリットとメリット

図1の要求は実世界の話なので，最初は人間が自然言語で話すか書くかしたものであろう，そして最後のコードはプログラミング言語である．途中は自然言語とかUML，形式仕様記述モデルなどになるだろう．

もし途中のものが存在するとすれば[注1]ほとんどが自然言語で書かれた文章で，いきなりソース・コードだけになっている例もあるかもしれない．その場合，中間の文書やモデルのようなものは，頭の中に存在していたということになる．また，最初は文書として存在していたが，保守されていないのでただ存在するだけになっているかもしれない．中間のものも，ソース・コードが保守されるように，役に立つ形式になるよう，工夫した方がよい．これが中間のものでシミュレーションや検証ができればソース・コードの保守の役に立つので放置されないのではないか．どんな形式がよいだろうか．

● 概要は自然言語，そのほかは形式表現

たとえば“function”という英語がある．日本語にすると，機能から関数まであり，その関数になると，数学で使う関数からプログラミング言語で使う関数まで意味が広がってしまう．この意味の広がりは詳細かつ正確に記述しなければならない．そうしないと曖昧さにつながってしまう．しかし，システム構想書など全く新しい話を始める場合には

注1：実際は何もないことがある．

KEYWORD――仕様，要求，自然言語，IEEE 830，CBMC