仕様記述からプログラム・コードの
自動生成，実機による実行まで

FM3マイコン基板で学ぶ Bメソッドによる形式手法

第2回 論理的な仕様を記述するために抽象機械を作成する

澤田 拓哉，浅野 誉寛

ソフトウェアの大規模化につれ，仕様通りの製品ができない，バグが複雑化する，そもそも仕様段階でおかしいことが製品になってから発覚したなどということが頻発しています．そこで，仕様段階から論理的に「正しい」ことを保証し，それを次第にコードへと落としていく形式手法が注目を集めています．

今回は，インターフェース仕様を記述するための「抽象機械」について解説します．

（編集部）

証明によって正しさを検証した仕様から，仕様を正しく満たすプログラム・コードを得るための方法論を形式方法といいます．Bメソッドは形式手法の一つで，連載第1回目（2012年7月号）で概要を紹介しました．

今回から自動販売機を制御するプログラムを題材に，無償の「Atelier B」という統合開発環境を使って実際に開発を進めながら重要事項を説明していきます．最終的に，正しさを検証した仕様から仕様を正しく満たすプログラムを作成し，本誌2012年6月号に付属のARM Cortex-M3マイコン搭載のFM3基板で動かしてみる予定です．

今回は，「抽象機械」の書き方を説明します．従来は仕様を文書などで記述していました．しかし文章による記述では表現があいまいであったり，不正確であったりすることがままあります．そのようなあいまいさを排除できるのが，今回説明する「抽象機械」のメリットです（**図1**）．

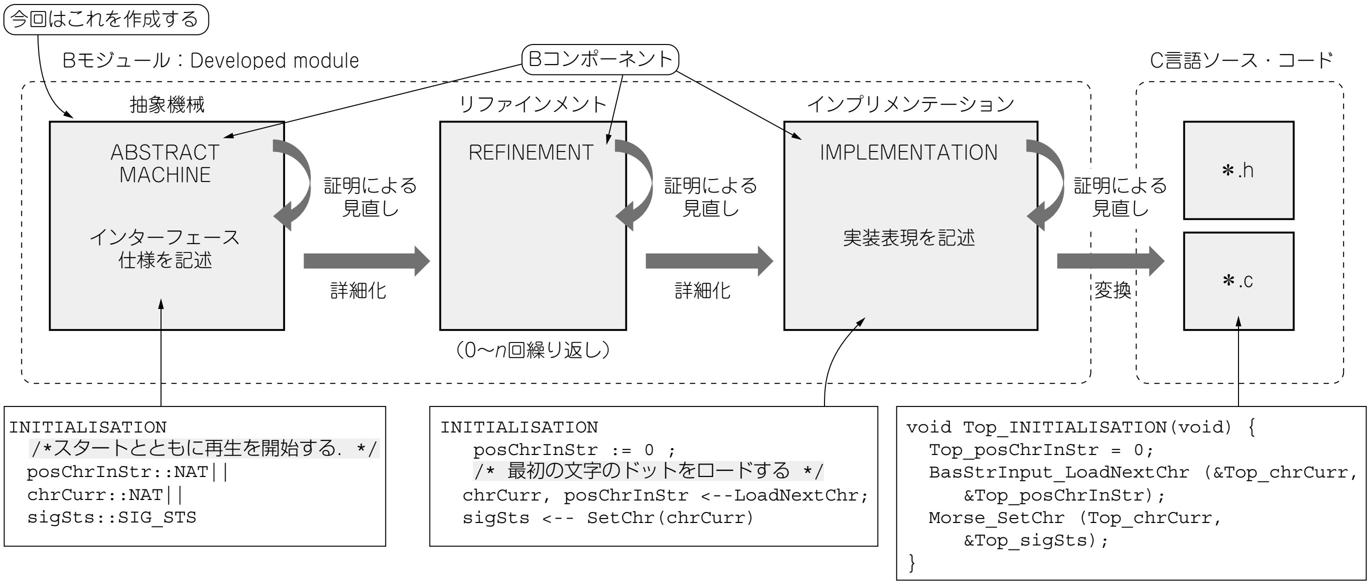
1 復習：Bメソッドを使った開発の流れ

● 全体は３ステップで開発完了

Bメソッドでは，

仕様＝「何を実現するか？」（What）
実装＝「どのように実現するか？」（How）

を明確に区別します（第1回参照）．例えば「ソートする」というプログラムの仕様と実装は**図2**のような関係になりま



```
INITIALISATION
  /*スタートとともに再生を開始する．*/
  posChrInStr::NAT||
  chrCurr::NAT||
  sigSts::SIG_STS
```

```
INITIALISATION
   posChrInStr := 0 ;
   /* 最初の文字のドットをロードする */
  chrCurr, posChrInStr <--LoadNextChr;
  sigSts <-- SetChr(chrCurr)
```

```
void Top_INITIALISATION(void) {
  Top_posChrInStr = 0;
  BasStrInput_LoadNextChr (&Top_chrCurr,
     &Top_posChrInStr);
  Morse_SetChr (Top_chrCurr,
     &Top_sigSts);
}
```

図1　形式手法の一つBメソッドのコンポーネントの種類と開発手順（再掲）
抽象機械からインプリメンテーションを作成する作業を詳細化という．実際には一度でインプリメンテーションできないので，間にリファインメントという作業が入る

注1：ただし，仕様と実装の間の隔たりが小さい場合は不要．

KEYWORD――形式手法，Bメソッド，Atelier B