



(3) 状態モデル

状態モデルとプログラムの対比は以下のようになる(図4参照)。

状態モデルにおける状態名はメソッド、振る舞いはそのメソッド内の振る舞いを表現している。メソッドの引数は通信モデルのイベントデータを参照し設定する。また通信モデルのイベントデータとメソッド名を一致させる。

状態1は外部クラスから呼び出されるため public となり状態2は内部からの呼び出しなので private となる。

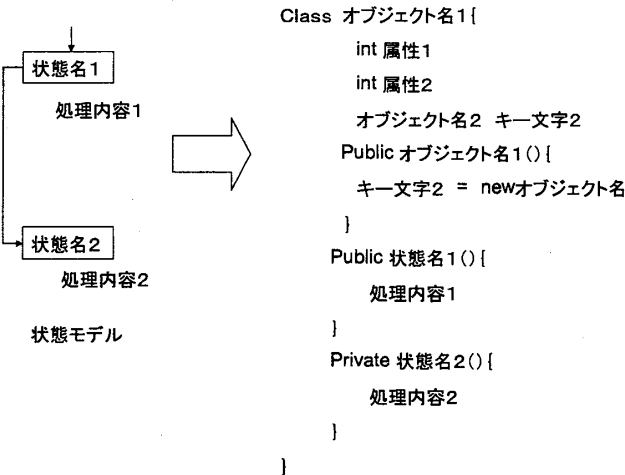


図4. 状態モデルとプログラムとの対応

3.3 結果

分析結果からプログラムへ変換する際、各工程で以下の問題点が生じる。

(1) 情報モデル

プログラムでは、情報モデルで欠落した変数などが存在する。情報モデルだけでは、プログラムの実行制御するために格納する変数、繰り返し回数や、処理に必要な変数などの抽出は困難である。

(2) 通信モデル

大きな問題は発見できなかった。

(3) 状態モデル

プログラムでは if 文を用いることにより、2つの状態を1つにまとめるため、分析結果とプログラムの状態数が異なる。

3.4 解決策

問題点を解決するために、各工程での解決策として以下のことが考えられる。

(1) 情報モデル

この段階で変数すべて抽出するのは困難である。このため、通信モデルや状態モデルの作成により変数を抽出して、情報モデルへフィードバックする。

(2) 通信モデル

verifier<sup>[4]</sup>(作成したモデルを動作させて、モデルの検証を行うツール)を使いシステムが動かないか確認を行う。

(3) 状態モデル

if 文を用いることにより 2つの状態を1つにまとめられるが、変更容易性や拡張性がなくなるため、1つにまとめず、分析結果をプログラムへ変換を行う。

4. 変換プロセス

分析方法から実装までを以下のように行う(図5参照)。

(1) S&Mによる分析

S&M法によって、対象システムの分析を行う。状態モデルでは各状態が何をするかをまとめる。

(2) 情報モデルの改善

状態、通信モデルの分析により必要になった変数を情報モデルに追加する。

(3) 通信モデルの改善

verifier を行いプログラムが動かないなどの問題が起きたら通信モデルの変更を行う。

(4) 状態モデルの改善

無駄な状態や必要な状態が抽出されたら変更し、最適化を行う。

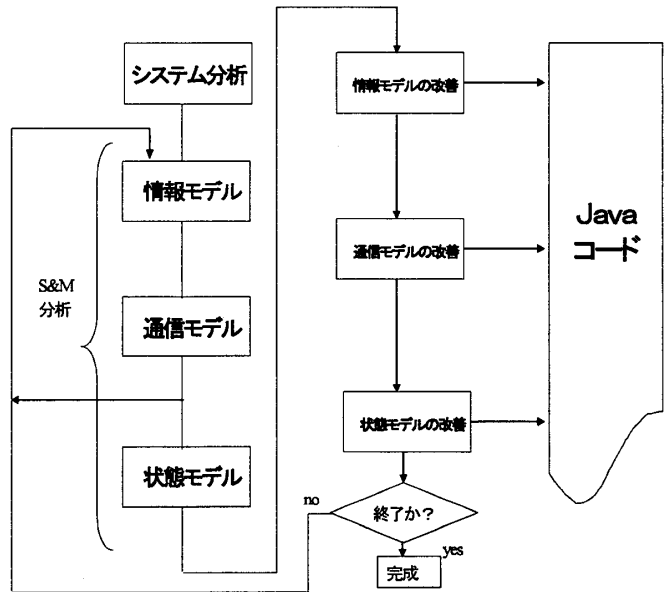


図5. 分析からプログラムへの変換プロセス

5. 今後の課題

小規模な単純なシステムでの見通しを得た。今後は大規模システムや継承、インターフェイス、マルチスレッド等を用いるシステムへの適応を行う。

6. 謝辞

本研究を進めるに当たり、ご助言・ご協力を頂きました株式会社東陽テクニカの二上貴夫部長に感謝いたします。

7. 参考文献

[1] Sally Shlaer/Stephen. J. Mellor  
「続オブジェクト指向システム分析」近代科学社 1995年  
[2] レオンスター「オブジェクト・モデリング」  
株式会社ピアソン/エデュケーション 1998年  
[3] UML Forum-Tokyo 2002 HOME  
HP: <http://www.omg.org/UMLForum2002/>  
[4] 東陽テクニカ HP: <http://www.toyo.co.jp/>